



THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **NANO-ELECTRONIQUE ET NANO-TECHNOLOGIES**

Arrêté ministériel : 7 août 2006

Et de

DOCTEUR DE L'UNIVERSITÉ DE MONASTIR

Spécialité : **PHYSIQUE**

Présentée par

« **Ahmed CHEFI** »

Thèse dirigée par « **Gilles SICARD** » et

codirigée par « **Rached TOURKI** »

préparée au sein du **Laboratoire TIMA**

dans l'**École Doctorale EEATS**

et du **Laboratoire EμE**

dans l'**École Doctorale Matériaux, Dispositifs et Microsystèmes**

Conception d'un micro-capteur d'image CMOS à faible consommation d'énergie pour les réseaux de capteurs sans fil

Thèse soutenue publiquement le «**28/01/2014**», à 10h30
devant le jury composé de :

M. Salvador MIR

Directeur de recherche au CNRS, Président

M. Dominique GINHAC

Professeur à l'Université de Bourgogne, Rapporteur

M. Mohamed MASMOUDI

Professeur à l'Université de Sfax, Rapporteur

M. Mohamed ATRI

Maître de conférences à l'Université de Monastir, examinateur

M. Gilles SICARD

Maître de conférences à l'Université Joseph Fourier, Directeur de thèse

M. Rached TOURKI

Professeur à l'Université de Monastir, Co-directeur de thèse

M. Adel SOUDANI

Maître de conférences, King Saud University, Encadrant, Invité



Remerciements

Je tiens à remercier très vivement mes encadrants Adel SOUDANI et Gilles SICARD qui m'ont accueilli, accompagné et conseillé tout au long de ce parcours. Mon envie de persévérer dans l'enseignement et la recherche doit beaucoup à l'énergie qu'ils savent mettre quotidiennement dans ces deux tâches.

Je tiens également à remercier Monsieur Rached TOURKI et Madame Dominique BORRIONE, de m'avoir accueilli dans leurs laboratoires respectifs et de m'avoir fourni tous les moyens nécessaires au bon déroulement de mon projet de recherche.

Mes plus sincères remerciements vont à Messieurs les professeurs Mohamed MASMOUDI et Dominique GINHAC qui m'ont fait l'honneur d'être les rapporteurs de cette thèse, à Monsieur Mahamed ATRI qui a accepté de participer à mon jury de soutenance ainsi qu'à Monsieur le professeur Salvador MIR qui a accepté de la présider.

Je remercie aussi l'Agence Universitaire de la Francophonie (AUF), la région Rhône-Alpes, le CNRS et le ministère de l'enseignement supérieur et de la recherche scientifique Tunisien, pour avoir financé cette thèse.

Enfin, je remercie très chaleureusement tous mes proches, ma famille et mes amis pour leur soutien et leurs encouragements.

RÉSUMÉ

Ce travail de recherche vise à concevoir un système de vision à faible consommation d'énergie pour les réseaux de capteurs sans fil. L'imageur en question doit respecter les contraintes spécifiques des applications multimédias pour les réseaux de capteurs de vision sans fil. En effet, de par sa nature, une application multimédia impose un traitement intensif au niveau du nœud et un nombre considérable de paquets à échanger à travers le lien radio, et par conséquent beaucoup d'énergie à consommer. Une solution évidente pour diminuer la quantité de données transmise, et donc la durée de vie du réseau, est de compresser les images avant de les transmettre. Néanmoins, les contraintes strictes des nœuds du réseau rendent inefficace en pratique l'exécution des algorithmes de compression standards (JPEG, JPEG2000, MJPEG, MPEG, H264, etc.). Le système de vision à concevoir doit donc intégrer des techniques de compression d'image à la fois efficaces et à faible complexité. Une attention particulière doit être prise en compte en vue de satisfaire au mieux le compromis "*Consommation énergétique - Qualité de Service (QoS)*".

ABSTRACT

This research aims to develop a vision system with low energy consumption for Wireless Sensor Networks (WSNs). The imager in question must meet the specific requirements of multimedia applications for Wireless Vision Sensor Networks. Indeed, a multimedia application requires intensive computation at the node and a considerable number of packets to be exchanged through the transceiver, and therefore consumes a lot of energy. An obvious solution to reduce the amount of transmitted data is to compress the images before sending them over WSN nodes. However, the severe constraints of nodes make ineffective in practice the implementation of standard compression algorithms (JPEG, JPEG2000, MJPEG, MPEG, H264, etc.). Desired vision system must integrate image compression techniques that are both effective and with low-complexity. Particular attention should be taken into consideration in order to best satisfy the compromise "Energy Consumption - Quality of Service (QoS)".

TABLE DES MATIERES

Introduction Générale.....	1
Chapitre 1: Les Réseaux de Capteurs Sans Fil : évolution vers les réseaux de capteurs de vision....	5
Introduction.....	5
1.1. Les réseaux de capteurs sans fil.....	5
1.1.1. Architecture typique d'un nœud de réseau de capteurs sans fil.....	7
1.1.2. Exemples d'applications des réseaux de capteurs sans fil.....	11
1.1.3. Spécificités et caractéristiques communes des réseaux de capteurs sans fil.....	12
1.1.4. Techniques d'optimisation de la consommation d'énergie pour les RCSFs.....	13
1.2. Les réseaux de capteurs de vision : analyse de l'existant et défis du futur.....	18
1.3. Traitement et transmission d'images à travers les réseaux de capteurs sans fil : analyse de l'existant et défis du futur.....	21
1.3.1. Applications potentielles des réseaux de capteurs d'image.....	22
1.3.2. Etat de l'art sur le traitement et la transmission d'images dans les réseaux de capteurs sans fil.....	23
1.3.2.1. Compression à la source.....	24
1.3.2.2. Compression distribuée d'images.....	28
1.3.2.3. Capteurs d'image conçus spécifiquement pour les RCSFs.....	30
1.4. Conclusion.....	32
Chapitre 2: Capteurs d'image CMOS pour les systèmes fortement contraints en énergie.....	34
Introduction.....	34
2.1. Les systèmes de vision CMOS.....	34
2.1.1. La matrice d'éléments photosensibles.....	35
A. Les photodétecteurs.....	35
B. Les pixels.....	38
2.1.2. Amplificateur colonne.....	41
2.1.3. Convertisseurs Analogiques Numériques (CAN) dédiés aux imageurs CMOS.....	41
2.1.4. Décodeurs d'adresse.....	44
2.1.5. Caractéristiques générales des imageurs CMOS.....	45
2.1.5.1. Dynamique.....	45
2.1.5.2. Courant d'obscurité.....	45
2.1.5.3. Rendement quantique.....	45
2.1.5.4. Facteur de conversion.....	45
2.1.5.5. Vitesse de lecture.....	46
2.1.5.6. Bruit dans les capteurs d'image CMOS.....	46
A. Bruit temporel.....	46
B. Bruit Spatial Fixe (Fixed Pattern Noise).....	47
2.1.5.7. Compensation du bruit dans les capteurs d'image CMOS.....	48
A. La technique CDS (Correlated Double Sampling).....	48
B. La technique NCDS (Non Correlated Double Sampling).....	49
C. La technique de double échantillonnage Delta (Double Delta Sampling).....	49

2.2.	Bilan énergétique et techniques d'optimisation pour les capteurs d'image CMOS.....	50
2.2.1.	Puissance dissipée par la matrice de pixels.....	50
2.2.2.	Puissance dissipée dans les décodeurs d'adresse.....	53
2.2.3.	Puissance dissipée dans les circuits analogiques "Front End " (amplificateur colonne).....	54
2.2.4.	Puissance nécessaire pour la conversion Analogique / Numérique.....	54
2.2.5.	Techniques d'optimisation de la puissance dissipée dans les imageurs CMOS.....	55
2.2.5.1.	Optimisations liées aux choix technologiques.....	55
2.2.5.2.	Optimisations aux niveaux algorithmique et architectural.....	56
2.2.5.3.	Optimisations aux niveaux circuit et logique.....	56
A.	Contrôle des courants de fuite.....	57
B.	Conception de circuits opérationnels à faible tension.....	57
C.	Polarisation du substrat.....	57
2.3.	Capteurs d'image CMOS intelligents pour les systèmes à ressources limités.....	58
2.4.	Conclusion.....	69

Chapitre 3: Techniques de compression d'images et de vidéos adaptées aux réseaux de capteurs de vision sans fil..... 70

Introduction.....	70
3.1. Nécessité de la compression pour les réseaux de capteurs de vision sans fil.....	71
3.2. Approches de compression d'image pour les réseaux de capteurs de vision sans fil.....	73
3.2.1. Techniques basées sur la transformée en cosinus discrète (DCT).....	74
3.2.2. Techniques basées sur la transformée en ondelettes discrète.....	81
3.3. Approches de compression vidéo pour les réseaux de capteurs de vision sans fil.....	93
3.4. Conclusion.....	99

Chapitre 4: De l'implémentation matérielle du système de vision vers l'implémentation du réseau de capteurs de vision sans fil idéal..... 100

Introduction.....	100
4.1. Architecture du système de vision.....	100
4.1.1. Le capteur d'image CMOS.....	100
4.1.2. Conversion Analogique/Numérique.....	103
4.1.3. Codeurs pour la compression d'image.....	103
A. Codeur à base de la transformée en ondelettes de Haar, le codage EZW et le codage RLE.....	105
B. Codeur à base de la transformée en ondelettes de Haar, le codage EZW et le codage arithmétique modifié.....	107
4.1.4. Codeur pour la compression vidéo.....	110
4.2. Vers l'implémentation d'un réseau de capteurs de vision sans fil à ultra-faible consommation d'énergie.....	112
4.2.1. Optimisations liées au matériel.....	113
4.2.2. Optimisations au niveau routage.....	116
4.3. Conclusion.....	117

Conclusion Générale et Perspectives.....	118
LISTE DES REFERENCES.....	121
LISTE DES PUBLICATIONS.....	127

TABLE DES FIGURES

CHAPITRE 1

Figure 1.1 : Schéma d'un réseau de capteurs sans fil.....	6
Figure 1.2 : Architecture matérielle d'un nœud de RCSF.....	7
Figure 1.3 : Topologie hiérarchique.....	14
Figure 1.4 : Schéma d'un réseau de capteurs de vision typique.....	22
Figure 1.5 : différentes formes de DCT zonale avec $k = 4$	26
Figure 1.6 : relation parent-enfant entre les différents coefficients d'ondelettes.....	27
Figure 1.7 : Politique d'écartement des paquets selon l'ordre de priorité.....	28
Figure 1.8 : Exemple de plusieurs capteurs de vision capturant des images corrélées.....	29
Figure 1.9 : Architecture du capteur de vision proposée.....	30

CHAPITRE 2

Figure 2.1 : Schéma bloc d'un système de vision CMOS classique.....	35
Figure 2.2 : Caractéristiques électriques de la photodiode.....	36
Figure 2.3 : Caractéristiques électriques d'un phototransistor.....	37
Figure 2.4 : Différents types de capteurs d'image CMOS : à pixels passifs (a) et à pixels actifs (b).....	38
Figure 2.5 : Pixel à intégration (a) et pixel à fonctionnement continu (b).....	39
Figure 2.6 : Architecture du pixel numérique (DPS).....	40
Figure 2.7 : Structure d'un amplificateur colonne standard.....	41
Figure 2.8 : Structure d'un convertisseur flash.....	42
Figure 2.9 : Structure d'un convertisseur pipeline N étages 1 bit.....	43
Figure 2.10 : Synoptique d'un convertisseur Analogique/Numérique Sigma-Delta.....	43
Figure 2.11 : Synoptique d'un modulateur Sigma-Delta du 1 ^{er} ordre.....	43
Figure 2.12: Synoptique d'un convertisseur Analogique/Numérique simple rampe.....	44
Figure 2.13: Illustration du Bruit Spatial Fixe.....	48
Figure 2.14: Technique de double échantillonnage corrélé (CDS).....	48
Figure 2.15: Technique de double échantillonnage non corrélé (NCDS).....	49
Figure 2.16 : Pixel et amplificateur colonne standards.....	50
Figure 2.17 : Imageur CMOS intégrant 7 processeurs SPIHT au pied de la colonne.....	60
Figure 2.18 : Imageur CMOS intégrant un codeur à base de la DCT-2D.....	61

Figure 2.19 : Topologie 4-connexe.....	65
Figure 2.20 : Transformation en ondelettes de Haar modifiée.....	66
Figure 2.21 : Architecture de l'imageur présenté dans [86].....	68

CHAPITRE 3

Figure 3.1 : Consommation énergétique du nœud source (plateforme MICA2) lors du traitement et de la transmission d'une image entière sans compression (image de taille 128x128 pixels codés sur 8 bpp).....	71
Figure 3.2 : Organigramme d'un Codeur/Décodeur basé sur la transformée en cosinus discrète.....	74
Figure 3.3 : Graphe des flux de la DCT de Loeffler.....	75
Figure 3.4 : Matrice de quantification JPEG pour la luminance.....	76
Figure 3.5 : Chaines de compression d'image à base de la transformée en cosinus discrète.....	77
Figure 3.6 : Caractéristique "débit-distorsion" des codeurs étudiés.....	78
Figure 3.7 : Consommations typiques d'un nœud source (plateforme MICA2) pour des applications liées au traitement et à la transmission d'images (de taille 128 x 128 pixels codés sur 8 bpp) pour un facteur de qualité $Q = 97$	79
Figure 3.8 : Distorsion des images reconstruites lors de l'application des chaines de compression étudiées.....	80
Figure 3.9 : Décomposition en ondelettes pour deux niveaux de décomposition.....	81
Figure 3.10 : Organigramme d'un Codeur/Décodeur basé sur la transformée en ondelettes.....	82
Figure 3.11 : Quantifieur scalaire uniforme à zone morte.....	84
Figure 3.12 : Exploitation des coefficients d'ondelettes par l'algorithme EZW.....	85
Figure 3.13. Déroulement des deux étapes de l'algorithme EZW pour deux seuils.....	87
Figure 3.14 : Chaîne de compression d'image à base de la transformée en ondelettes de Haar, le codage EZW et le codage RLE.....	87
Figure 3.15 : Caractéristique "débit-distorsion" du codeur étudié.....	88
Figure 3.16 : Consommation énergétique d'un nœud de RCSF de type MICA2 lors de la compression d'une image (128x128 pixels – 8 bpp) et la transmission du code résultant.....	88
Figure 3.17 : Consommation énergétique d'un nœud de RCSF de type MICA2 lors du codage (en utilisant le codage arithmétique) d'une image transformée (128x128 pixels – 8 bpp).....	90
Figure 3.18 : Chaîne de compression d'image à base de la transformée en ondelettes de Haar, le codage EZW et une version améliorée du codage arithmétique.....	90
Figure 3.19 : Caractéristique "débit-distorsion" du codeur étudié.....	91
Figure 3.20 : énergie consommée par le nœud capteur (plateforme <i>Mica2</i>) lors de la compression (codeur de la figure 3.18) d'une image (128x128 pixels codés sur 8 bpp) puis la transmission du code résultant.....	92
Figure 3.21 : Système de vision pour les réseaux de capteurs sans fil.....	94
Figure 3.22 : Algorithme de compression vidéo proposé.....	96

Figure 3.23 : Caractéristique débit-distorsion du codeur (blocs de taille 8x8 pixels – 8bpp) et consommations typiques du nœud (plateforme MICA2) lors de l’implémentation du codeur..... 98

Figure 3.24 : Caractéristique débit-distorsion du codeur (blocs de taille 16x16 pixels – 8bpp) et consommations typiques du nœud (plateforme MICA2) lors de l’implémentation du codeur..... 98

CHAPITRE 4

Figure 4.1 : Layout d’un pixel standard..... 101

Figure 4.2 : Architecture globale du système de vision adoptée..... 101

Figure 4.3 : Pixel standard et amplificateur colonne..... 103

Figure 4.4 : Schéma simplifié de la circuiterie de calcul à base de capacités commutées..... 104

Figure 4.5 : Schéma bloc du codeur étudié (DWT-Quantification-EZW-RLE)..... 105

Figure 4.6 : Schéma bloc du codeur étudié (DWT-Quantification-EZW-Arithmétique modifié)..... 107

Figure 4.7 : Schéma bloc du codeur vidéo proposé..... 110

Figure 4.8 : Modèle d’un réseau de capteurs de vision sans fil..... 113

Figure 4.9 : Consommations typiques des nœuds de deux plateformes de réseaux de capteurs sans fil..... 115

Figure 4.10 : Consommations énergétiques dues aux protocoles de routage lors de la transmission d’une image non compressée (image de taille 128x128 pixels codés sur 8 bpp)..... 116

LISTE DES TABLEAUX

CHAPITRE 1

Tableau 1.1 : Consommations typiques des microcontrôleurs TI msp430x2xx et Atmel ATmega164P à 3 V et 4 MHz.....	13
Tableau 1.2 : Variation de la vitesse et de la consommation énergétique en fonction de la tension d'alimentation.....	16
Tableau 1.3 : Comparaison entre quelques plateformes de réseaux de capteurs de vision.....	20

CHAPITRE 2

Tableau 2.1 : Récapitulatif des différentes implémentations de CAN pour les capteurs d'image.....	42
---	----

CHAPITRE 3

Tableau 3.1 : Énergie consommée par différents codeurs vidéo implémentés sur la plateforme STARGATE (plateforme de réseaux de capteurs sans fil).....	97
---	----

CHAPITRE 4

Tableau 4.1 : Performances du codeur étudié : implémentation sur une carte FPGA Xilinx Virtex 5.....	106
Tableau 4.2 : Performances du codeur étudié : implémentations ASIC pour différentes technologies standards d'intégration.....	106
Tableau 4.3 : Performances du codeur étudié : implémentation sur une carte FPGA Xilinx Virtex 5.....	107
Tableau 4.4 : Performances du codeur étudié : implémentations ASIC pour différentes technologies standards d'intégration.....	108
Tableau 4.5 : Résultats d'implémentations du codeur JPEG 2000 en technologie CMOS standard 0.18 μm	109
Tableau 4.6 : Performances du codeur vidéo : implémentations ASIC pour différentes technologies standards d'intégration.....	111
Tableau 4.7 : Performances de quelques standards de compression vidéo implémentés en technologie CMOS standard 0.35 μm	112
Tableau 4.8 : Consommations typiques de quelques transceivers de réseaux de capteurs sans fil.....	114
Tableau 4.9 : Consommations typiques de quelques microcontrôleurs utilisés au sein des nœuds de réseaux de capteurs sans fil.....	114
Tableau 4.10 : Consommations typiques lors de l'accès mémoire pour les plateformes de réseaux de capteurs sans fil, MICA et Telos.....	114

Introduction Générale

Les avancées récentes dans le domaine de la micro-électronique et des technologies de communication sans-fil ont permis la création d'une combinaison entre les systèmes embarqués et les systèmes distribués, ayant engendré les Réseaux de Capteurs Sans-Fil (RCSFs). Les capteurs apparaissent comme des systèmes autonomes miniaturisés, équipés d'un processeur (ou d'un microcontrôleur) embarqué, un *transcepteur* (appelé aussi *transceiver*) radio, une mémoire interne et une batterie. Les nœuds du réseau ont pour mission de récolter des données et les transmettre à un nœud spécifique nommé « station de base ».

Les caractéristiques attirantes de ces systèmes (faible coût de développement, autonomie de fonctionnement, simplicité d'installation, etc.) ont ouvert la voie à une multitude d'applications constituant les sièges privilégiés de leur déploiement. Parmi toutes les applications potentielles des RCSFs, celles utilisant des capteurs d'image sont appréciables puisqu'elles touchent à des domaines particulièrement sensibles (surveillance, micro-caméras pouvant transmettre des images de l'intérieur d'un corps humain, vidéosurveillance dans un secteur stratégique ou difficile d'accès, etc.). Toutefois, beaucoup d'obstacles inhérents à leurs spécificités doivent être surmontés avant de pouvoir atteindre leur maturité technique. Parmi ces entraves, le problème de limitation de ressources (énergie, puissance de calcul, espace mémoire, débit de transmission, etc.) se pose avec acuité, en raison des contraintes de miniaturisation et de coût de fabrication, et doit être solutionné de manière efficace et en conformité avec les caractéristiques particulières des RCSFs. En plus, les plateformes de capture d'image existantes sur le marché (CMUCam3, Cyclops, OmniVision, etc.) sont loin de répondre aux besoins particuliers des RCSFs (faible consommation d'énergie, miniaturisation du dispositif de capture de l'image, difficultés liées à l'interfaçage du capteur d'image avec les nœuds du RCSF approprié, etc.). Heureusement, les avancées technologiques en micro-électronique ont permis le développement de micro-caméras embarquées dans pratiquement tous les dispositifs électroniques existants (téléphones cellulaires, ordinateurs portables, assistants numériques personnels, RCSFs), sans augmentation significative du coût de l'équipement (coût énergétique, prix, poids, forme, etc.). La technologie CMOS est ici privilégiée car, grâce à sa consommation extrêmement réduite, elle permet de concevoir des circuits intégrés mixtes qui répondent aux exigences des applications multimédias dédiées aux RCSFs.

Dans ce contexte, ce travail de recherche vise à concevoir un capteur d'image CMOS miniaturisé et à faible consommation d'énergie, afin d'être embarqué au sein des nœuds du RCSF approprié. Certes, le développement d'un seul dispositif électronique intégrant à la fois le système de vision et le nœud standard de RCSFs résout une partie du problème (réduction de la consommation d'énergie liée à l'acquisition de l'image et une communication aisée entre le nœud et la caméra), mais la solution est loin de faire face aux défis majeurs à lever. En effet, de par sa nature, une application multimédia impose un traitement intensif au niveau du nœud et un nombre considérable de paquets à transmettre sur le lien radio, et donc beaucoup d'énergie à consommer. Une solution évidente pour diminuer la quantité de données transmises, et donc la durée de vie du réseau, est de compresser l'image ou la vidéo à la source. Néanmoins, les contraintes strictes des nœuds du réseau rendent inefficace en pratique l'exécution des algorithmes de compression standards (JPEG, JPEG2000, MJPEG, MPEG, H264, etc.). La plupart des travaux de recherche liés aux RCSFs admettent que le *transcepteur* radio est le composant le plus gourmand en énergie. En fait, l'énergie nécessaire pour transmettre un seul bit peut être suffisante pour effectuer des centaines voire des milliers d'instructions (le rapport varie en fonction de la plateforme utilisée). Malgré l'important écart de consommation, la plupart des techniques de compression efficaces ont une complexité algorithmique très élevée, pouvant aboutir à des consommations énergétiques considérables. Une attention particulière doit donc être prise en vue de satisfaire au mieux le compromis *Complexité Algorithmique*, *Taux de Compression* et *Qualité de Service (QoS)*. L'enjeu est donc d'opter pour une technique de compression permettant d'atteindre un taux de compression élevé avec une faible complexité de calcul et une bonne qualité d'image à la réception.

Rappelons que, Grâce aux technologies d'intégration CMOS submicroniques dédiées aux applications à faible consommation d'énergie, nous pensons qu'il est possible de concevoir un capteur d'image intelligent intégrant des fonctionnalités de compression sur le plan focal de l'imageur, et pouvant fonctionner en tant que coprocesseur embarqué sur le nœud du RCSF. L'objectif de la thèse est de proposer un capteur d'image CMOS intelligent, intégrant une méthode de compression efficace satisfaisant aux contraintes posées par les RCSFs, c'est-à-dire ayant une faible complexité algorithmique, un taux de compression élevé et une bonne qualité d'image à la réception.

Organisation du document

Ce manuscrit de thèse est organisé en 4 chapitres. Le positionnement de nos travaux sera présenté sur les deux premiers chapitres et nos contributions seront détaillées dans les deux derniers.

Le premier chapitre présente de façon générale les *réseaux de capteurs sans fil*, puis détaille le cas particulier des *réseaux de capteurs de vision sans fil*. Ce chapitre traite aussi les contraintes des *réseaux de capteurs d'image* actuels pour des applications liées au traitement et au transfert d'images, notamment la consommation élevée d'énergie. La dernière partie de ce chapitre sera réservée à la présentation des techniques d'économie d'énergie pour les réseaux de capteurs de vision sans fil, et plus particulièrement la compression de données. L'état de l'art concernant les travaux de recherche liés à la compression et la transmission de données à travers ce type de réseaux sera présenté également.

Une introduction aux capteurs d'image sera présentée dans le deuxième chapitre. Vu les caractéristiques attirantes de la technologie d'intégration CMOS, les capteurs d'image CMOS seront étudiés en détail. Nous allons montrer dans cette partie du manuscrit l'intérêt des systèmes de vision CMOS et leur aptitude à répondre aux besoins spécifiques des réseaux de capteurs de vision existants (économie d'énergie, facile à embarquer au sein d'un nœud standard de RCSF, possibilité d'intégrer du traitement sur le plan focal de l'imageur, etc.). La dernière section de ce chapitre détaille l'état de l'art des systèmes de vision intégrant des techniques de conservation de l'énergie ; le cas particulier de la compression d'image sera traité en détail, vu la nécessité impérieuse de réduire les flux de données émanant d'une application multimédia.

La dernière partie du manuscrit, composée par les chapitres 3 et 4, permet de positionner nos contributions par rapport aux travaux de recherche menés au niveau international, et permet d'évaluer les résultats de performance des approches proposées. Le chapitre 3 permet de justifier, tout d'abord, la nécessité de la compression pour les réseaux de capteurs de vision sans fil (consommations élevées si on n'applique aucune technique de compression). La deuxième partie de ce chapitre détaille les implémentations logicielles, au sein des nœuds sources du réseau, des algorithmes de compression d'image et de vidéo choisis ; les résultats obtenus ont permis d'atteindre des gains énergétiques considérables, à la fois pour les nœuds sources ainsi que les autres nœuds du réseau. Le quatrième et dernier

chapitre prouve l'intérêt des implémentations matérielles des codeurs étudiés par rapport aux implémentations logicielles (en termes de temps de calcul et d'énergie consommée). L'architecture du système de vision complet sera étudiée en détail dans cette partie du manuscrit.

Enfin, les différents points abordés dans ce manuscrit sont synthétisés dans une conclusion générale qui présente également quelques perspectives pour la suite de ce travail.

Chapitre 1

Les Réseaux de Capteurs Sans Fil : évolution vers les réseaux de capteurs de vision.

Introduction

Avec l'intégration des capteurs dans les systèmes embarqués, l'élargissement de l'éventail d'applications pour les réseaux de capteurs sans fil (RCSFs) est devenu possible. Au cours des dernières années, un nouveau domaine de recherche a attiré quelques groupes de chercheurs, motivés par ce nouveau défi : il s'agit des réseaux de capteurs de vision. En effet, certaines applications ont besoin d'identifier avec exactitude les objets qui traversent le champ surveillé par le RCSF [1]. L'information multimédia représente un support incontournable pour la validation de plusieurs applications tels que l'identification, la localisation et le suivi d'objets. Toutefois, le traitement multimédia se trouve confronté au problème de consommation exhaustive de l'énergie ce qui révèle la nécessité d'une solution spécifique pour les nœuds d'un réseau de capteurs multimédia [2, 3]. Ce chapitre fait une description générale des réseaux de capteurs sans fil, puis détaille et argumente l'évolution vers les réseaux de capteurs de vision, jugés prématurés d'un point de vue industriel pour l'instant.

1.1. Les réseaux de capteurs sans fil

De nombreuses avancées techniques dans les domaines des MOEMS (Micro-Opto-Electro-Mechanical Systems) et des technologies de communications sans fil, ont permis le développement de petits objets communicant équipés de capteurs à coût raisonnable. Ces objets, appelés nœuds, capteurs ou hôtes, intègrent une unité de calcul (microcontrôleur ou microprocesseur), un ou plusieurs dispositifs de capture (capteur de température, d'humidité, de pression, de fumée, de mouvement, etc.), de mémoires, d'une unité de communication sans fil et d'une batterie d'alimentation ou d'un système de récupération de l'énergie de l'environnement [4, 5].

Les RCSFs font partie des réseaux ad hoc (ou MANET, pour Mobile Ad hoc Network). Dans ce type de réseaux, les nœuds s'organisent seuls et de manière totalement décentralisée, formant ainsi un réseau autonome et dynamique ne reposant sur aucune infrastructure filaire. Les hôtes du réseau peuvent être fixes ou mobiles selon la nature de l'application [6].

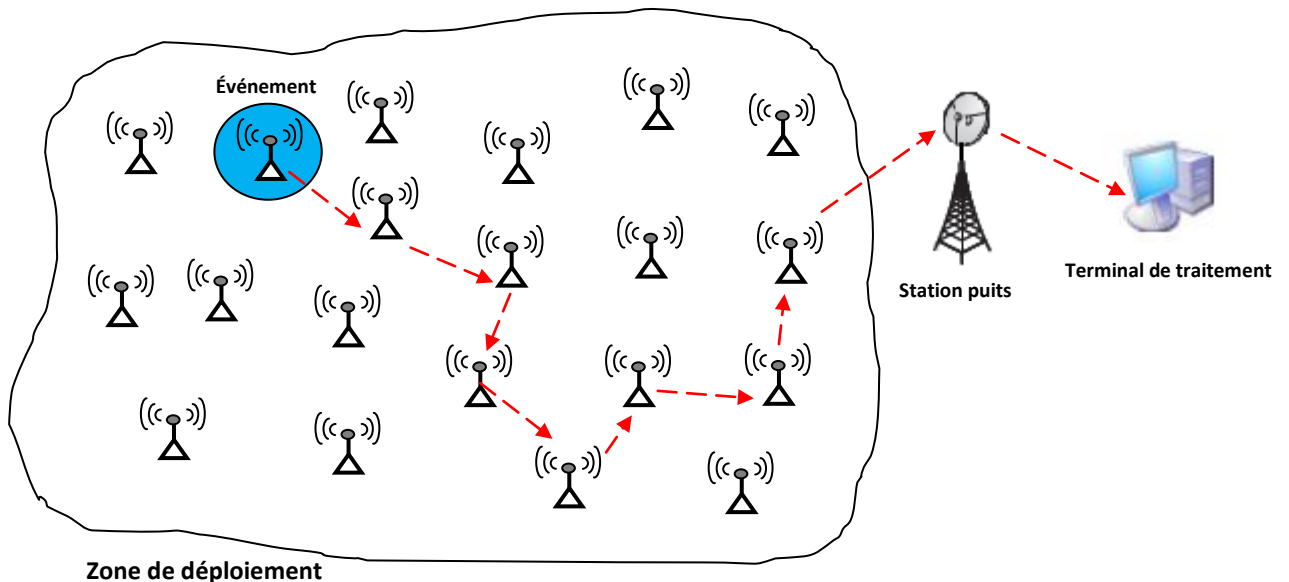


Figure 1.1 : Schéma d'un réseau de capteurs sans fil

Un exemple typique de réseau de capteurs sans fil est illustré par la figure 1.1 : les nœuds capteurs ou hôtes sont disséminés de manière aléatoire sur toute la zone à contrôler. La station de base, généralement appelée puits et située à l'extrémité de la zone à surveiller, est chargée de collecter les informations en provenance des capteurs et de les transférer vers un terminal de traitement qui analyse ces données. Dans ce cas de figure, lorsqu'un événement se produit (brusque variation de température ou de pression par exemple), une alerte est acheminée par le biais d'une communication multi-sauts vers le puits. Il s'agit donc de collecte d'informations suite à un événement [6]. Il existe deux autres méthodes pour récupérer les données du réseau : collection à la demande et collection périodique. La première méthode consiste à émettre une demande par le nœud puits vers tous les nœuds du réseau (broadcast) pour qu'ils remontent leurs derniers rapports vers le puits. En ce qui concerne la collection périodique, les capteurs prennent des mesures (température, pression, etc.) dans des intervalles de temps réguliers [4, 7].

1.1.1. Architecture typique d'un nœud de réseau de capteurs sans fil

Le schéma d'une architecture typique d'un nœud capteur est présenté sur la figure 1.2. Le but de cette partie est de détailler le fonctionnement de chaque bloc fonctionnel du nœud capteur.

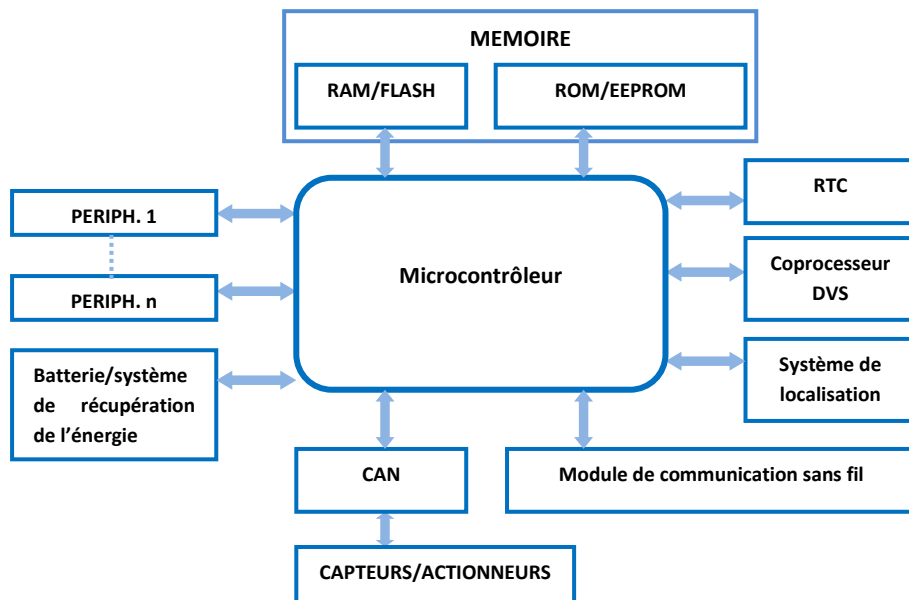


Figure 1.2 : Architecture matérielle d'un nœud de RCSF

- Microcontrôleur

Le microcontrôleur représente le cœur du nœud du RCSF. Il reçoit les données provenant des capteurs, traite ces données, décide quand et où les envoyer et gère la commande des actionneurs (s'ils existent). Le microcontrôleur reçoit aussi des paquets en provenance des nœuds voisins, les analyse et répond à ces messages, si nécessaire. D'autres tâches plus complexes sont confiées au microcontrôleur telles que la gestion de la mémoire, la gestion de l'énergie (migrer du mode actif au mode veille ou veille prolongée et vice versa), ordonnancement des tâches, gestion des périphériques, etc. Les plateformes existantes de RCSFs utilisent une grande variété de microcontrôleurs à architectures différentes [8]. Nous vous présenterons ci-après les microcontrôleurs les plus employés par les constructeurs des RCSFs.

Parmi les microcontrôleurs les plus répondus (dédiés pour les RCSFs), on trouve le *MSP430* de Texas Instrument. Il s'agit d'un microcontrôleur 16 bits réputé pour sa faible consommation d'énergie. Le *MSP430* est intégré dans de nombreuses plateformes de réseaux de capteurs telles que *Telos*, *BSN*, *Wavenis* et *eyesIFX*. Deux familles très faible consommation sont disponibles : la famille des *MSP430F1xx*, ayant une puissance dynamique

inférieure mais fonctionne à une cadence plus faible (ne dépasse pas 8 MHz), et la famille des *MSP430F2xx* qui peut atteindre une fréquence de fonctionnement de 16 MHz [1].

Intel StrongARM est un processeur RISC 32 bits qui peut atteindre une fréquence de fonctionnement de 206 MHz, mais il est utilisé dans des applications nécessitant de traiter des flux importants de données (applications vidéos, applications temps-réel, etc.) [7, 8].

Atmel ATmega164P est un microcontrôleur RISC 8 bits. Il consomme presque 2 fois l'énergie consommée par le *MSP430x2xx* pour tous les modes de fonctionnement.

Mica est un microcontrôleur universitaire asynchrone QDI 8 bits CISC. Il a été conçu au laboratoire TIMA avec la collaboration de France Télécom R&D, et fabriqué par ST-Microelectronics en technologie 0,25 μm avec une bibliothèque de cellules standards. Le microcontrôleur dispose d'un bus de données de 8 bits, un bus d'adressage de 16 bits, une mémoire RAM de 16 Ko, un ROM de 2 Ko et un périphérique permettant de gérer jusqu'à 6 ports parallèle de 8 bits et deux ports série. La consommation du microcontrôleur dépend de son mode de fonctionnement (varie de 0.8 mW sous 1 V à 77 mW sous 3.5 V) [1].

Le *Lutonium* est un microcontrôleur universitaire 8051 asynchrone conçu en technologie asynchrone quasi-insensible aux délais au *California Institute of Technology*, et fabriqué en technologie CMOS 0,18 μm de TSMC. La consommation du microcontrôleur varie en fonction du mode de fonctionnement (de 0.17 mW sous 0.5 V à 100 mW sous 1.8 V) [1].

- Convertisseur Analogique / Numérique (CAN)

Un Convertisseur Analogique/Numérique (CAN) est nécessaire pour convertir les signaux analogiques provenant des capteurs en valeurs numériques exploitables par le microcontrôleur. Les différentes techniques de conversion *Analogique/Numérique* seront détaillées dans le chapitre suivant (section 2.1.3). Le choix d'un CAN dépend des besoins de l'utilisateur et des contraintes imposées par l'application (vitesse de conversion, coût du convertisseur, énergie consommée par conversion, surface occupée par le CAN, etc.). Les bénéfices en termes de réduction d'énergie des circuits asynchrones comparativement aux circuits synchrones ont poussé les chercheurs à concevoir des CAN asynchrones (un CAN asynchrone prend un échantillon uniquement lorsque la grandeur physique varie de plus d'un quantum d'amplitude prédéfini) [1, 8]. La technologie asynchrone souffre cependant à l'heure

actuelle d'un manque de maturité qui peut expliquer la rareté des CAN asynchrones commercialisés.

- Horloge temps réel

Elle utilise souvent un quartz piézoélectrique afin de conserver l'information de temps dans le système. Les protocoles de communication se servent de l'horloge temps réel pour avoir une heure commune entre les nœuds (synchroniser les hôtes entre eux). Cette horloge peut aussi servir au système logiciel en tant que *Timer* pour réveiller le microcontrôleur à des dates bien précises.

Comme cette partie matérielle est active en permanence, sa consommation énergétique doit donc être faible pour avoir une longue durée de vie du nœud. La consommation typique des horloges temps réel est de quelques centaines de nano-ampères, ce qui rend sa consommation comparable aux consommations en veille des autres blocs fonctionnels du nœud [1].

- Module de communication sans fil

Ce bloc représente la partie matérielle qui consomme le plus d'énergie. Le module de communication sans fil peut être un émetteur/récepteur optique ou à ultrasons, mais généralement il s'agit d'un émetteur/récepteur radio qui utilise principalement l'une des 3 bandes ISM (Industriel, Scientifique et Medical) suivantes : la bande des 400 MHz, la bande des 900 MHz et la bande des 2.4 GHz. Plusieurs travaux de recherche ont permis de réduire la consommation du module radio afin de prolonger la durée de vie du nœud capteur. La plupart de ces études ont visé la réduction du phénomène de l'écoute passive ou l'écoute inutile ; mais malgré cela, la consommation due à ce phénomène reste importante (la consommation lors de l'écoute passive est du même ordre de grandeur que celle due à l'émission et la réception de données) [1, 8].

- Système de localisation

Un capteur n'est utile que si une information relative à sa localisation géographique est fournie. Le GPS (Global Positioning System) est le système de localisation le plus connu et utilisé au niveau mondial. D'autres systèmes de géolocalisation existent ; parmi les quels on cite le système russe GLONASS, le système IRNSS indien, le système Beidou développé par la chine et le système de localisation européen Galileo. Le principe général de ces technologies se décrit comme suit : lorsqu'un module de localisation demande sa position, chaque satellite lui communique ses coordonnées et la distance qui le sépare de lui. Au moins

3 satellites sont nécessaires pour une localisation dans un espace à deux dimensions, alors qu'au moins 4 satellites le sont pour un espace à trois dimensions [9].

- Capteurs/actionneurs

L'unité de détection contient un ou plusieurs capteurs embarqués sur le nœud. Ces capteurs transforment le ou les grandeurs physiques observées (pression, température, humidité, vitesse, mouvement, etc.) en grandeur électrique (tension électrique, intensité, train d'impulsions, etc.). Certaines plateformes de réseaux de capteurs sont composées de nœuds équipés d'actionneurs. Un actionneur permet de convertir une énergie d'entrée (généralement une énergie électrique dans le cas des RCSFs) en une énergie de sortie (généralement un signal électrique de commande) utilisable pour commander ou contrôler un processus (par exemple déclencher une alarme suite à un incendie) [1, 10].

- Mémoires

Les nœuds capteurs sont équipés de plusieurs mémoires de type RAM (Random Access Memory), ROM (Read Only Memory), EEPROM ou flash. La mémoire de programme permet de sauvegarder la suite des instructions du ou des programmes à exécuter par le nœud. La mémoire de données permet le stockage des données à traiter par le nœud capteur. Une mémoire de configuration est figée au sein du nœud (généralement de type ROM ou EEPROM), elle maintient les données de configuration nécessaires au bon fonctionnement de l'ensemble des blocs fonctionnels de l'hôte. La taille et le type de ces mémoires varient en fonction de la plateforme utilisée [1].

- Coprocasseur DVS

La technique DVS (Dynamic Voltage Scaling), ou adaptation dynamique en tension, permet de contrôler efficacement la vitesse du microcontrôleur afin d'optimiser la consommation d'énergie. La technique permet de ralentir le processeur (lorsque cela est possible) en abaissant la tension d'alimentation. En effet, dans un circuit CMOS, l'énergie consommée est proportionnelle au carré de la tension d'alimentation alors que la vitesse du microcontrôleur varie linéairement en fonction de cette tension. Ainsi, une faible diminution de la tension V_{dd} ralentit faiblement le circuit mais permet de réduire de manière importante la consommation énergétique [1, 8].

- Batterie / système de récupération de l'énergie

Les nœuds disposent d'une batterie d'alimentation comme source principale d'énergie. Certaines plateformes utilisent des systèmes de récupération de l'énergie de l'environnement.

De nos jours, l'utilisation des matériaux piézo-électriques est le moyen de récupération de l'énergie le plus adéquat pour les nœuds de RCSFs [11].

1.1.2. Exemples d'applications des réseaux de capteurs sans fil

Les applications les plus populaires des RCSFs utilisent des capteurs qui permettent de mesurer de la température, de l'humidité, de l'acoustique, des vibrations, de la pression, etc. Ces applications représentent le standard classique d'applications dédiées pour ce type de réseaux.

Nous détaillons dans ce qui suit d'autres applications récentes, plus évoluées et plus importantes [7, 8, 11] :

- *Surveillance des machines industrielles*

L'idée est de fixer des capteurs sur des endroits difficiles d'accès afin de détecter des événements qui indiquent le besoin de maintenance (vibration, fumée, bruits et nuisances sonores, etc.).

- *Détection d'incendies*

Les nœuds capteurs, déployés dans une forêt par exemple, produisent collectivement une " carte de température " de toute la surface couverte. Pour ce genre d'applications, le coût des capteurs doit être suffisamment faible pour pouvoir déployer un grand nombre de nœuds et couvrir une surface plus grande.

- *Le bâtiment*

L'installation d'un RCSF à l'intérieur d'un bâtiment permet de remédier efficacement au problème de perte d'énergie (mauvaise ventilation, mauvais usage d'air conditionné, etc.). Les nœuds capteurs sont utilisés généralement pour un meilleur contrôle de la température et de l'humidité, ce qui augmente le niveau de confort des habitants. Les RCSFs sont déployés aussi dans les régions à haute activité sismique pour contrôler la tenue aux contraintes mécaniques.

- *Applications militaires*

Les premières applications potentielles des réseaux de capteurs sans fil ont concerné le domaine militaire. L'idée est de déployer des nœuds nanoscopiques, et donc invisibles, sur un champ de bataille pour surveiller les mouvements des ennemis.

- Applications médicales

Les RCSFs peuvent être utilisés pour surveiller et contrôler les fonctions vitales d'un être humain. Des micro-capteurs avalés ou implantés sous la peau peuvent traiter certaines maladies sans recours à la chirurgie (détection précoce d'un cancer, implantation d'une rétine artificielle pour corriger la vue, etc.).

1.1.3. Spécificités et caractéristiques communes des réseaux de capteurs sans fil

Dans cette partie, nous allons dégager toutes les caractéristiques communes aux réseaux de capteurs sans fil, indépendamment du type de l'application et de la plateforme utilisée [11, 12, 13].

- L'auto-organisation

L'auto-organisation fait référence à l'organisation des hôtes du réseau sans interaction avec une entité extérieure et sans contrôle centralisée. Des interactions locales entre les nœuds capteurs permettent au réseau de trouver et maintenir une structure qui répond de manière efficace aux besoins de l'application. Un changement local ne doit que faiblement impacter la structure globale.

- Collaboration

Parfois, un nœud est incapable de décider tout seul si un événement s'est produit ou non. Il doit donc coopérer avec les nœuds voisins pour prendre la bonne décision.

- La tolérance aux fautes

La défaillance d'un ou de quelques nœuds ne doit pas entraîner la défaillance de tout le réseau. Le RCSF doit être capable de fonctionner partiellement et avec des performances réduites au lieu de tomber complètement en panne.

- Notion de qualité de service (QoS : Quality Of Service)

On parle de *QoS* lorsqu'on souhaite atteindre un comportement meilleur du réseau. Les paramètres qui influent le critère *QoS* sont les suivants : le taux de pertes de paquets, la gigue (différence de délai de transfert de bout en bout entre des paquets d'un même flux de données), le débit (en kbits/sec) et l'atténuation du signal. L'atténuation du signal est fortement liée à la distance qui sépare les nœuds. Ce paramètre n'affecte pas trop la qualité de service puisque les RCSFs utilisent le plus souvent des communications multi-sauts.

- La flexibilité

Les nœuds capteurs doivent être capables de s'adapter rapidement et efficacement en fonction de la situation du réseau (mettre à jour la table de routage suite à la défaillance d'un nœud, diminuer le critère QoS quand l'énergie disponible devient critique, etc.).

- Scalabilité

La surveillance d'un phénomène peut nécessiter le déploiement d'un nombre de nœuds qui est de l'ordre de plusieurs centaines voire plusieurs milliers de capteurs. Le réseau doit pouvoir exploiter la nature fortement dense de certaines applications.

1.1.4. Techniques d'optimisation de la consommation d'énergie pour les RCSFs

Il existe de nombreux mécanismes pour réduire la consommation d'énergie dans les RCSFs, tels que les modes faible consommation, l'adaptation dynamique des tensions d'alimentation, les protocoles de communication spécifiques, la compression de données, etc. Ces mécanismes sont dans la plupart du temps des mécanismes matériels gérés par le logiciel de plus haut niveau. Nous allons détailler dans cette partie les principes de ces mécanismes.

- Modes faible consommation

Lorsqu'un circuit synchrone n'a aucune tâche à exécuter, il peut se mettre en veille, en attendant une interruption qui le réveille. L'absence des modes faible consommation entraîne le fonctionnement continu de l'horloge interne même s'il n'y a aucun traitement à réaliser, ce qui génère une consommation supplémentaire inutile. Ces modes sont contrôlés par le logiciel et permettent de couper les différentes horloges du circuit pendant qu'il attend une interruption [1].

Microcontrôleur	Consommations typiques
TI msp430x2xx	Mode actif : 1200 μ J Mode veille profonde : 0.1 μ J Retour vers le mode actif : 18 nJ
Atmel ATmega164P	Mode actif : 2100 μ J Mode veille profonde : 0.2 μ J Retour vers le mode actif : 36 nJ

Tableau 1.1 : Consommations typiques des microcontrôleurs TI msp430x2xx et Atmel ATmega164P à 3 V et 4 MHz

Les écarts de consommation entre le mode actif et le mode veille profonde des deux microcontrôleurs sont très conséquents et justifient l'utilisation de ces modes. En effet, les rapports de consommations sont de 12 000 pour le *TI msp430x2xx* et de 10 500 pour *Atmel*

ATmega164P. En plus, les deux microcontrôleurs consomment très peu d'énergie pour passer du mode veille prolongée au mode actif.

- Protocoles de routage spécifiques

Le critère de base pour la conception d'un protocole de routage dédié aux réseaux de capteurs sans fil est son efficacité à économiser l'énergie. Parmi les protocoles de routage les plus connus et les plus efficaces, on cite le protocole *LEACH* (*Low Energy Adaptive Clustering Hierarchy*) qui est un protocole de routage hiérarchique (Un protocole hiérarchique consiste à découper le réseau en plusieurs régions; chaque région communique avec le reste du réseau à travers un ou plusieurs nœuds spécifiques appelés *Cluster-Heads*). L'idée est de former des groupes ou clusters de nœuds capteurs en se basant sur la puissance du signal reçu, et d'employer les *Cluster-Heads* pour acheminer les messages locaux vers la station de base [14, 15].

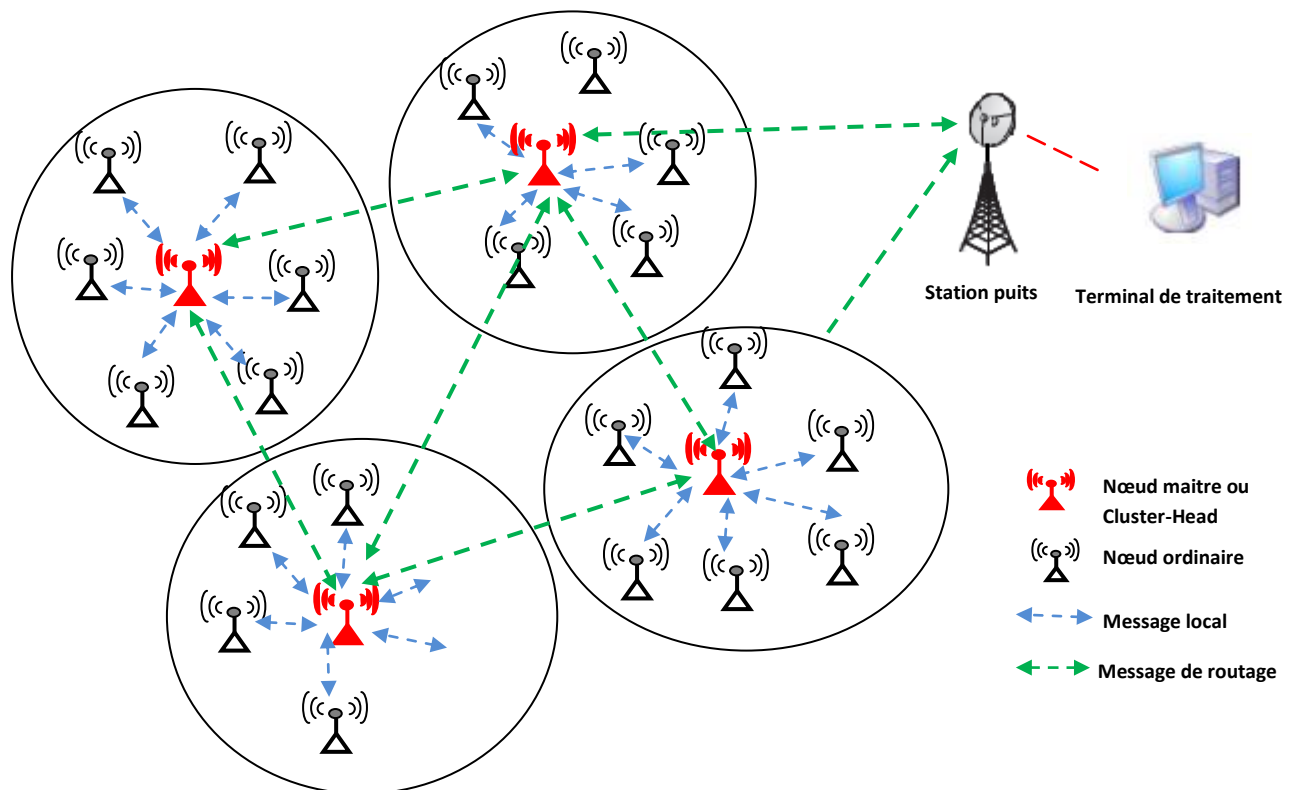


Figure 1.3 : Topologie hiérarchique

Cette topologie permet d'économiser de l'énergie puisque seuls les *Cluster-Heads* effectuent une transmission vers la station de base ou nœud puits. En plus, pour chaque nœud ordinaire un *TimeSlot* est alloué, utilisé pour échanger les données avec le *Cluster-Head*, et le nœud se met hors tension pendant les autres *TimeSlots* pour économiser de l'énergie. L'élection des *Cluster-Heads* se fait périodiquement pour ne pas épuiser rapidement les

batteries de certains nœuds par rapport à d'autres. D'autres protocoles de routage existent dans la littérature (SPIN, SPEED, Directed Diffusion, Flooding, TEEN, APTEEN, etc.). Le choix du protocole de routage adéquat dépend de plusieurs paramètres dont la nature de l'application, la densité du réseau, la plateforme utilisée, la qualité de service demandée, etc [4, 8, 16].

- Protocole de communication ZigBee

Les contraintes des RCSFs et en particulier celle portant sur la consommation d'énergie montrent clairement que les standards sans fil classiques (Bluetooth, WLAN, etc.) ne sont pas adaptés à ce type de réseaux. Le standard ZigBee a été conçu pour faire face à ces contraintes. Il permet des communications radio à faible coût énergétique entre les nœuds capteurs [17]. C'est un protocole à bas débit caractérisé par une portée comprise entre quelques mètres et quelques centaines de mètres et un débit faible (Inférieur ou égal à 250kbits/s). ZigBee s'appuie sur le standard IEEE 802.15.4 pour les couches physique et liaison, qui sont les couches 1 et 2 du modèle *OSI* (*Open Systems Interconnections*: standard de communications en réseau) ainsi que sur le développement des couches réseaux et application par la ZigBee Alliance. Les nœuds ZigBee nécessitent entre 2% et 10% du code nécessaire à la mise en œuvre de nœuds Bluetooth. Cette caractéristique permettra aux nœuds ZigBee d'occuper moins d'espace mémoire et de consommer moins d'énergie par rapport aux réseaux sans fil classiques [17, 18].

- Adaptation dynamique de la tension d'alimentation

La technique *DVS* (*Dynamic Voltage Scaling*) permet de contrôler dynamiquement l'énergie consommée en fonction de l'activité du circuit. En effet, la consommation énergétique d'un circuit CMOS est proportionnelle au carrée de la tension d'alimentation V_{dd} et de la capacité totale commutée C (équation 1.1). La vitesse du circuit, quant à elle, varie peu en fonction de la tension V_{dd} [19].

$$P_{CMOS} = C \cdot V_{dd}^2 \cdot f \quad (\text{éq. 1.1})$$

$$f = \frac{1}{Ld \cdot td} \quad (\text{éq. 1.2})$$

$$td = k \cdot \frac{V_{dd}}{(V_{dd} - V_t)^2} \quad (\text{éq. 1.3})$$

P_{CMOS} : puissance consommée par le circuit CMOS

C : capacité totale commutée

f : fréquence de fonctionnement du circuit

V_{dd} : tension d'alimentation du circuit

V_t : tension de seuil dépendant de la technologie
 K : constante dépendant de la technologie
 L_d : longueur du chemin critique ou délai maximum
 td : délai ou retard de propagation d'un circuit

En se basant sur les équations détaillées ci-dessus, on voit bien qu'une faible diminution de la tension d'alimentation ralentit un peu le circuit mais permet de réduire considérablement l'énergie consommée [8, 20, 21]. Le tableau 1.2 justifie l'intérêt de la technique de l'adaptation dynamique de la tension d'alimentation. Les gains énergétiques varient en fonction de la plateforme et la technologie utilisées. Pour le msp430F1xx, une chute de 45% de la vitesse du microcontrôleur sera compensée par un gain en puissance de 85%. Concernant le microcontrôleur *Lutonium*, le gain en puissance peut dépasser 99%, mais il est accompagné d'une chute considérable de la vitesse du circuit (baisse de 98%).

Tension (V)	Vitesse maximale	Consommation (mW)	Energie par cycle (nJ)	Energie par instruction
Microcontrôleur msp430F1xx				
3.3	7.3 MHz	11.4	1.56	1.16 à 9.69 nJ
2.2	5 MHz	3	0.6	0.6 à 3.6 nJ
1.8	4.15 MHz	1.6	0.39	0.39 à 2.34 nJ
Microcontrôleur msp430F2xx				
3.3	16 MHz	22.6	1.41	1.41 à 8.46 nJ
2.2	7.5 MHz	4.5	0.6	0.6 à 3.6 nJ
1.8	4.15 MHz	1.6	0.39	0.39 à 2.34 nJ
Microcontrôleur Lutonium				
1.8	200 MIPS	100	ND	0.5 nJ
0.9	66 MIPS	9.2	ND	0.139 nJ
0.5	4 MIPS	0.17	ND	0.043 nJ
Processeur SNAP				
1.8	240 MIPS	ND	ND	218 pJ
0.9	61 MIPS	ND	ND	55 pJ
0.6	28 MIPS	ND	ND	25 pJ

Tableau 1.2 : Variation de la vitesse et de la consommation énergétique en fonction de la tension d'alimentation [1]

- Systèmes d'exploitation spécifiques

Au niveau logiciel, l'utilisation d'un système d'exploitation dédié aux RCSFs (très fortement contraints en énergie et en mémoire) permet de réduire la consommation d'énergie de manière significative. Parmi les "*OS*" (*Operating Systems*) destinés aux RCSFs, on cite Contiki, SOS, Think et TinyOS. Ce dernier, développé à l'université de Berkeley, est le plus

utilisé vu qu'il occupe un espace mémoire très réduit et il consomme une énergie ultra faible en fonctionnement [18, 22].

TinyOS est un OS évènementiel qui ne possède pas d'espace *kernel-user* (dans un OS classique, l'espace *kernel* est réservée à l'exécution des tâches liées au cœur du système d'exploitation alors que l'espace *user* est dédiée aux applications lancées par l'utilisateur) et qui ne permet ni allocation dynamique, ni mémoire virtuelle. TinyOS a donc une empreinte mémoire très faible puisqu'il n'occupe que 300 à 400 octets dans le cadre d'une distribution minimale. TinyOS s'appuie sur le langage NesC, une extension du langage C, qui propose une architecture basée sur des composants qui sont assemblés statiquement à la compilation et permettant de réduire efficacement la taille mémoire de l'OS ainsi que ses applications. NesC permet de déclarer deux types de composants : les modules et les configurations. Les modules constituent les briques élémentaires de code et implantent une ou plusieurs interfaces. À plus haut niveau, les configurations sont des composants permettant de faire le lien entre les différents composants (modules ou configurations), et de connecter les interfaces de ces composants entre elles [23, 24, 25].

Chaque composant correspond à un élément matériel (LEDs, Timer, CAN, Radio) et peut être réutilisé dans différentes applications. Une application se définit comme une multitude de composants associés ensemble. L'implantation de composants s'effectue en déclarant des tâches, des commandes ou des évènements, qui communiquent entre eux à travers des interfaces.

Les tâches sont employées pour exécuter les activités de longue durée. Une file d'attente (FIFO: First In First Out) permet d'organiser l'exécution des tâches. Il n'y a pas de mécanisme de préemption entre les tâches et une tâche activée s'exécute en entier. Lorsque la file d'attente est vide, le système d'exploitation met en veille le nœud capteur jusqu'à l'apparition d'une nouvelle tâche à exécuter.

Les commandes permettent de lancer l'exécution d'une fonctionnalité implantée dans un autre composant (il s'agit d'un appel de fonctions). De même, les évènements sont aussi des appels de fonctions, mais celles-ci ont pour but de signaler qu'un évènement, tel que la fin d'un calcul, s'est produit. L'apparition d'une interruption matérielle engendre l'exécution instantanée d'un évènement NesC, prioritaire par rapport aux tâches et qui peut interrompre la tâche en cours d'exécution [23].

- Optimisation de la consommation d'énergie par la couche MAC

La couche MAC (*Media Access Control*) ou couche d'accès au support est la couche logicielle qui permet d'organiser et contrôler l'accès multiple au medium (ou canal de communication). Les protocoles MAC fondés sur la méthode TDMA (*Time Division Multiple Access*) offrent une solution implicite : chaque nœud du réseau n'échange ses messages qu'à des intervalles de temps (slots) qui lui sont réservés ; il doit garder son transcepteur éteint durant le reste des slots pour économiser de l'énergie [10].

- Compression de données

Comme le *tranceiver* (ou transcepteur) radio est le composant le plus gourmand en énergie dans un nœud de RCSF, il est bien clair que le fait de réduire la quantité de données à transmettre permet d'économiser de l'énergie de manière efficace. La solution la plus évidente est la compression de données, mais parfois les techniques de compression de données utilisées ont un coût énergétique très élevée, surtout lorsqu'il s'agit de compresser des images ou des vidéos. Il faut donc développer des algorithmes de compression qui soient peu gourmands en énergie et qui aient un bon rapport *débit-distorsion* [7, 8].

Donc, de nos jours, les défis majeurs posés sont la compression et la transmission d'images et de vidéos à travers les RCSFs. Les applications qui font appel à l'acquisition et au traitement d'image nécessitent des précautions spécifiques pour faire face aux contraintes strictes de ce type de réseaux. Ce problème sera abordé plus en détail tout au long de ce manuscrit.

1.2. Les réseaux de capteurs de vision : analyse de l'existant et défis du futur

Les difficultés liées aux systèmes de vision sont généralement associées à des problèmes de traitements à coûts relativement élevés. Ces problèmes pourraient être multipliés lorsque nous devons faire face à d'énormes limitations en ressources, ce qui est le cas des RCSFs [26, 27]. Au-delà des défis traditionnels de ce type de réseaux, les applications multimédias posent des défis particuliers :

- Nécessité de validation expérimentale des techniques proposées

La plupart des approches proposées en littérature sont évaluées soit par analyse mathématique, soit par simulation. Ces études considèrent des hypothèses assez simplistes concernant la topologie du réseau, les techniques de communication entre les nœuds, les

caractéristiques des nœuds, etc. Les modèles théoriques et les hypothèses proposés doivent donc être validés sur des plates-formes réelles [7].

- Absence de capteurs de vision adéquats pour les RCSFs

La plupart des capteurs d'image existants sur le marché sont destinés à être intégrés dans des dispositifs dotés de ressources importantes en termes de mémoire, énergie et vitesse de calcul (ordinateurs et téléphones portables, assistants numériques personnels, etc.). Les dispositifs de capture d'image disponibles actuellement sont donc loin de répondre aux exigences strictes des RCSFs [28].

Le tableau 1.3 présente les caractéristiques générales des systèmes de vision les plus connus sur le marché. Ces plateformes ont été exploitées par quelques équipes de recherche dans le domaine des réseaux de capteurs de vision sans fil. Les travaux effectués ne permettent pas d'avoir une comparaison objective entre ces plateformes. En effet, l'absence du même environnement de travail (mêmes *testbenches*, même taille d'image, même débit binaire ou *bitrate*, mêmes caractéristiques du RCSF utilisé, etc.) rend difficile la comparaison entre ces systèmes de vision.

Nous remarquons ici que la plupart des plateformes de vision utilisent un processeur 32-bit (à l'exception de Cyclops et WiCa qui utilisent un processeur 8-bit). En fait, les applications qui font appel à des traitements intensifs d'images nécessitent des processeurs performants pour accélérer l'exécution des tâches. Une convolution 2D effectuée par un processeur 8-bit fonctionnant à 4 MHz nécessite un temps d'exécution 16 fois plus élevé qu'un processeur ARM7 32-bit cadencé à 48 MHz. Ce dernier consomme uniquement une énergie 6 fois plus élevée par rapport au premier, mais pour des applications à contraintes énergétiques sévères, l'utilisation d'un processeur 8-bit semblerait plus efficace [8, 28].

Plateforme	Imageur/Resolution	Processeur	Trancepteur/ Fréq/Bitrate	Consommations typiques
Cyclops (2005)	ADCM-1700 352x288	ATmega128L 8-bit RISC	CC1000 de MICA2, 0.9 GHz, 76.8 kbps	211.8 mW (résolution 128x128), 2.6 fps, 4.97 μ J/pixel
Imote2-based (2006)	OV7649 640x480	PXA271 32-bit ARM5	CC2420 d'Imote2, 2.4 GHz, 250 kbps	322 mW (résolution 80x60), 8 fps, 8.38 μ J/pixel
MeshEye TM (2007)	ADNS-3060 (2) 30x30 ADCM-2700 640x480	AT91SAM7S 32-bit ARM7	built-in CC2420, 2.4 GHz, 250 kbps	155 mW (résolution 64x64), 0.5 fps, 75.68 μ J/pixel
WiCa (2007)	OM6802 (2) 640x480	Xetal-II+8051 SIMD PE array + 8-bit CISC	CC2420 d'AquisGrain, 2.4 GHz, 250 kbps	500 mW (résolution 320x120), 30 fps, 0.43 μ J/pixel
FireFly Mosaic (2007)	OV6620 352x288	LPC2106 32-bit ARM7	CC2420 de FireFly, 2.4 GHz, 250 kbps	114.5 mW (résolution 352x288), 5.2 fps, 0.27 μ J/pixel
CITRIC (2008)	OV9655 1280x1024	PXA270 32-bit ARM5	CC2420 de TELOSB 2.4 GHz, 250 kbps	970 mW (résolution 320x240), 3.3 fps, 3.83 μ J/pixel
EyeRIS TM -based (2010)	Q-Eye 176x144	Q-Eye + NIOS-II SIMD PE array + 32-bit 3ISC	RF230 de IRIS, 2.4 GHz, 250 kbps	1500 mW (résolution 176x144), 15 fps, 3.94 μ J/pixel

Tableau 1.3 : Comparaison entre quelques plateformes de réseaux de capteurs de vision [28]

La puissance consommée des plateformes présentées au tableau 1.3 varie entre 114.5 mW et 1500 mW. Ces consommations sont trop élevées, par conséquent ces systèmes de vision sont loin d'être appliqués sur les réseaux de capteurs sans fil fortement contraints en énergie. D'autres plateformes de vision existent ; parmi lesquels on cite la CMUCam3 composée d'une caméra CMOS Omnicision OV6620 ou une OV7620, un microcontrôleur NXP LPC2106 avec 64 Ko de RAM, 128 Ko de mémoire flash et un FIFO de 1 Mo pour le stockage des flux vidéo. Ce dispositif peut être connecté à un nœud de RCSF de type Telos. La CMUCam3 n'est pas conçu pour des applications à faible consommation d'énergie et donc elle n'est pas appréciable dans le cas des RCSFs. Nous citons aussi la plateforme de capture de vidéo *Panoptes*, qui consomme environ 5 W en mode capture vidéo et 58 mW en mode endormi (mode sleep). La plateforme utilise une architecture logicielle intégrée sur un noyau

Linux embarqué et intègre la norme de compression JPEG, mais sa consommation élevée rend impossible son utilisation avec les RCSFs [7].

- *Compression d'image sur les réseaux de capteurs de vision*

Puisque le transcepteur radio est le bloc le plus gourmand en énergie dans un nœud capteur, la compression des données à transmettre semblerait le moyen le plus efficace pour économiser de l'énergie. Ce raisonnement n'est pas toujours vrai. En effet, la plupart des techniques de compression d'images existantes ont une complexité algorithmique considérable et par conséquent ils sont très gourmands en énergie [29]. L'algorithme de compression choisi doit bénéficier à la fois d'un bon rapport *débit-distorsion* et d'une consommation énergétique optimale. L'algorithme doit aussi occuper un espace mémoire raisonnable par rapport aux ressources disponibles sur le nœud capteur [30]. L'état de l'art concernant la compression d'image sur les réseaux de capteurs de vision sera détaillée par la suite tout au long de ce manuscrit.

- *Contraintes spécifiques aux réseaux de capteurs de vision*

Les travaux de recherche sur les images et les vidéos sont beaucoup plus complexes que les contributions qui traitent des signaux analogiques ou numériques simples (température, humidité, etc.). En effet, la détection et le traitement d'un signal simple tel que la température ambiante nécessite l'emploi de quelques octets ; par contre la capture et le codage d'une image conduit à l'emploi de plusieurs centaines voire plusieurs milliers d'octets. En plus, la complexité du matériel est multipliée : la capture de la température ou l'humidité dans un milieu quelconque nécessite un seul capteur, mais le fait de capturer une image nécessite plusieurs photo-capteurs (un photo-capteur par pixel). Cette évidence entraîne un coût supplémentaire en énergie (pour la capture, le traitement et la transmission de l'image si nécessaire), en mémoire, et en temps de capture et de traitement [31].

1.3. Traitement et transmission d'images à travers les réseaux de capteurs sans fil : analyse de l'existant et défis du futur

Malgré l'existence de plusieurs travaux de recherche concernant les réseaux de capteurs sans fil, peu de propositions ont été accordées au cas particulier de traitement et de transmission d'images à travers ce type de réseaux. La plupart des études étaient focalisées sur les protocoles de communication à tous les niveaux (optimisation de la consommation d'énergie, fiabilité, etc.). Quelques équipes de recherche ont essayé quand même de

confronter les difficultés liées aux applications multimédias pour les RCSFs. Dans cette section, nous allons tout d'abord mettre l'accent sur l'intérêt de ce type d'applications en citant quelques applications potentielles faisant intervenir le transfert d'images à travers les RCSFs. Ensuite, nous allons détailler l'état de l'art des différentes approches employées pour le traitement et le transfert d'images à travers les réseaux de capteurs de vision sans fil. La dernière partie de la section sera consacrée d'une part à discuter les limites des plateformes et des approches actuelles, et d'autre part à présenter notre vision sur les réseaux de capteurs d'images du futur.

1.3.1. Applications potentielles des réseaux de capteurs d'image

Dans cette sous-section, nous allons présenter quelques applications potentielles des RCSFs exigeant l'emploi de capteurs de vision (figure 1.4). On distingue des réseaux de capteurs de vision à un saut (*single-hop network*: communication directe entre deux nœuds) et des réseaux de capteurs de vision multi-sauts (*Multi-hop network*: la communication entre deux nœuds peut s'appuyer sur des nœuds intermédiaires). Pour des raisons de coût, généralement des nœuds spécifiques du réseau seront équipés par des capteurs de vision. Le reste des nœuds seront utilisés pour transmettre les images acquises jusqu'au nœud puits.

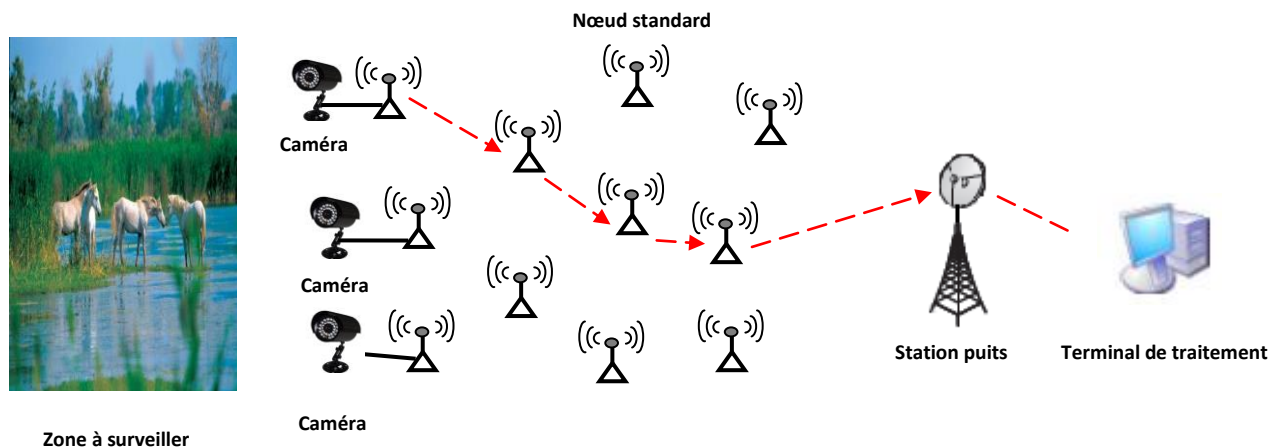


Figure 1.4 : Schéma d'un réseau de capteurs de vision typique

De nombreuses applications réelles des réseaux de capteurs de vision existent dans la littérature. Parmi ces applications, on cite :

- Suivi du trafic routier

Des réseaux de capteurs de vision peuvent être déployés pour contrôler la circulation routière. Des techniques de traitement et d'analyse d'images peuvent être employées pour

faire le dénombrement des véhicules afin d'estimer le niveau du trafic en fonction des heures de la journée.

- Sécurité des zones sensibles

Les réseaux de caméras sans fil peuvent être appliqués pour la sécurité des zones sensibles telles que les centrales nucléaires (détection de pannes ou d'intrusions), les zones sauvages (dénoncer des chasseurs illégaux), les zones industrielles, etc.

- Vigilance environnementale

Les nœuds peuvent être déployés dans des endroits stratégiques pendant plusieurs mois afin d'obtenir des images qui ont un intérêt scientifique. Par exemple, un réseau de capteurs infrarouges Cyclops a été installé dans la réserve des Montagnes *James San Jacinto* en *Californie* pour étudier la nidification d'une espèce particulière d'oiseaux. De même, un RCSF équipé de caméras a été déployé dans le parc *Ni'inlii Njike* au *Canada* pour étudier le comportement des *ours grizzly* [7].

1.3.2. Etat de l'art sur le traitement et la transmission d'images dans les réseaux de capteurs sans fil

Des études récentes ont montré que la transmission d'images entières sans compression à travers les RCSFs actuels reste encore très coûteuse en énergie [32, 33, 34]. Cristian Duran-Faundez et al ont mesuré dans [7] la consommation d'un nœud Mica2 de Crossbow lié à une caméra Cyclops lors de l'acquisition puis la transmission d'une image de taille 128x128 pixels (8 bits par pixel). Le nœud a consommé 91 mJ sans compter les en-têtes des paquets alors que la caméra avait une consommation énergétique d'environ 90 mJ. Ces consommations, jugées trop élevées pour les RCSFs, épuisent rapidement les batteries des nœuds [7, 35]. Vu le coût énergétique important du transfert d'images entières à travers le réseau, la compression devient donc une nécessité. Dans la littérature des RCSFs, plusieurs algorithmes de compression ont été proposés. Ces algorithmes peuvent être classés en deux catégories :

- Algorithmes locaux : un seul nœud effectue la compression puis transmet le code résultant à travers le reste du réseau. Il s'agit d'une compression à la source.
- Algorithmes distribués : vu la nature distribuée des RCSFs, le processus de compression est partagé entre plusieurs nœuds du réseau.

1.3.2.1. Compression à la source

Dans la littérature, quelques propositions privilégient le traitement local pour éviter la complexité de mise en œuvre d'une distribution de charge entre plusieurs nœuds du réseau. La compression à la source bénéficie de certains avantages. En effet, elle permet d'étendre la durée de vie du réseau ; une attention particulière doit être prise pour éviter d'épuiser rapidement la batterie du nœud source. La compression locale permet aussi de contribuer à la diminution de la congestion du réseau. En plus, des techniques spécifiques peuvent être implémentées localement (par exemple la technique de mélange des blocs) afin d'aboutir à une tolérance maximale aux pertes de paquets (les données perdues seront distribués sur toute l'image pour une meilleure qualité d'image à la réception).

Parmi les algorithmes implémentés, on cite *JPEG2000*, *JPEG*, *SPIHT*, *DCT* et *DWT*. D'autres techniques hybrides (algorithmes basés sur un mélange de différentes techniques de compression) ont été utilisées afin d'amener des économies importantes d'énergie. Le coût d'implémentation de ces techniques dépend de la plateforme utilisée.

Certains groupes de recherche ont utilisé des techniques basées sur la norme de compression *JPEG*. Le standard *JPEG* consiste à diviser l'image en blocs de $N \times N$ pixels (généralement des blocs de 8×8 ou 16×16) et appliquer sur chaque bloc : la transformée en cosinus discrète (*DCT: Discrete Cosine Transform*), une étape de quantification, le codage *RLE* (*Run Length Encoding*) et le codage de *Huffman* [36]. Parmi toutes les étapes citées précédemment, la *DCT* (*Discrete Cosine Transform*) est la plus coûteuse en nombre d'instructions. Le calcul de la *DCT* classique pour chaque coefficient du bloc à traiter se fait comme suit :

$$A_{i,j} = \frac{1}{4} \cdot C_i \cdot C_j \sum_{x=0}^7 \sum_{y=0}^7 x_{i,j} \cdot \cos\left(\frac{(2x+1)i\pi}{16}\right) \cdot \cos\left(\frac{(2y+1)j\pi}{16}\right)$$
$$\text{Avec } C_f = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } f = 0, \\ 1 & \text{si } f > 0, \end{cases} \quad 0 \leq i, j \leq 7$$

La complexité algorithmique de la *DCT* classique a poussé les chercheurs à essayer d'autres algorithmes qui donnent des résultats semblables à la *DCT* mais avec un nombre réduit d'instructions (11 multiplications et 29 additions uniquement pour la *DCT* de Loeffler contre 176 multiplications et 464 additions pour la *DCT* classique si on utilise un bloc de 8×8

pixels). Kaddachi et al [37] ont bénéficié des caractéristiques attirantes de la *DCT de Loeffler* (rapidité d'exécution et complexité algorithmique réduite) pour proposer une technique de compression à la source dédiée aux RCSFs. La chaîne de compression étudiée repose sur la *DCT de Loeffler*, un bloc de quantification utilisant la table de quantification JPEG, et enfin le *codage de Golomb* suivi par le *codeur MQ* (une variante du codage arithmétique utilisée dans le standard JPEG 2000). La technique de mélange de blocs ou de pixels est utilisée en amont de toute la chaîne de compression pour garantir une meilleure qualité d'image à la réception en cas de perte de paquets [7]. Parmi les techniques utilisées pour mieux résister aux pertes de paquets dans les RCSFs, on cite la technique des Automorphismes de Tore (AT) qui consiste à changer les positions des pixels dans une image en se basant sur le calcul suivant :

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^n \begin{pmatrix} i \\ j \end{pmatrix} \bmod N$$

Avec : (i, j) les coordonnées du pixel avant transformation

(i', j') les coordonnées du pixel après transformation

N : taille de l'image (image carrée de taille $N \times N$ pixels)

n : clé de diffusion ($n \in \mathbb{N}$)

k : valeur quelconque à choisir ($k \in \mathbb{N}$)

La technique de compression proposée permet d'atteindre des taux de compression élevés (atteignant 0.4 bit par pixel), mais une implémentation logicielle (le nœud capteur lui-même effectue la compression) n'est pas appréciable grâce à la complexité du traitement effectué. Pour cette raison, Kaddachi et al ont opté pour une implémentation matérielle (conception d'un circuit à ultra-faible consommation dédié au traitement d'image). Le circuit doit fonctionner comme co-processeur intégrée au sein du capteur lui-même. L'inconvénient de cette approche est le coût supplémentaire engendré par une éventuelle intégration de l'*IP* (*Intellectual Property*) développé au sein des nœuds du RCSF.

Makkaoui et al [38] ont prouvé l'efficacité de la *DCT zonale* pour une implémentation sur un nœud source de réseau de capteur sans fil. Cette technique permet de réduire considérablement la complexité algorithmique de la *DCT classique* sans dégrader significativement la qualité de l'image à la réception. La technique consiste à opérer sur un sous-ensemble de coefficients situés dans une zone du bloc de 8×8 pixels (il s'agit de sélectionner les pixels les plus significatifs situés en haut et à gauche de chaque bloc).

DC	4	2	0				
7	-3	-2	-1				
-1	1	-1	3				
2	0	2	1				

(a) Forme carrée

DC	4	2	0				
7	-3	-2					
-1	1						
2							

(b) Forme triangulaire

Figure 1.5 : différentes formes de DCT zonale avec $k = 4$

K est un coefficient qui délimite horizontalement et verticalement la zone de sélection des coefficients à traiter. Dans le cas de la figure 1.5 (a), on doit traiter uniquement k^2 coefficients (16 coefficients pour $k = 4$) au lieu de 64 coefficients pour la DCT classique. Si on utilise la forme triangulaire (figure 1.5 (b)), uniquement $\frac{1}{2} \cdot k \cdot (k + 1)$ coefficients doivent être traités (10 coefficients dans le cas où $k = 4$). Malgré la faible complexité de la DCT zonale par rapport à la DCT classique, la chaîne de compression complète (DCT zonale + quantification + codage entropique + paquetsation) possède une complexité considérable. Le gain énergétique atteint pourrait être amélioré par une implémentation matérielle de l'approche présentée.

Certaines équipes de recherche ont essayé d'implémenter l'algorithme de compression JPEG 2000 localement sur un nœud de RCSF. JPEG 2000 est basé sur des techniques assez complexes et assez gourmandes en espace mémoire, temps de calcul et consommation énergétique. La technique inclut une transformée en ondelettes, une étape de quantification, et un codage arithmétique adaptatif permettant d'atteindre un taux de compression élevé. Wu et Abouzeid ont proposé une technique basée sur le standard JPEG 2000, mais qui respecte les contraintes exigeantes des réseaux de capteurs de vision [39]. Ils ont introduit une *heuristique* appelé *MTE (Minimize Total Energy)* : le nœud source compresse l'image capturée puis calcule le taux de compression et l'énergie dissipée. Si le taux de compression cherché est atteint ou bien l'énergie dissipée dépasse un seuil prédéfini, on arrête le processus de compression et on transmet l'image codée. Si non, une nouvelle itération est réalisée et les

nouvelles valeurs de l'énergie dissipée et du taux de compression seront calculés et stockés. Cette méthode vise à optimiser la consommation des nœuds de transit, mais elle implique d'épuiser très rapidement la batterie du nœud source.

Wu et Chen ont proposé un schéma de compression basé sur le codage SPIHT pour les RCSFs [2, 40]. L'algorithme commence par appliquer la transformée en ondelettes discrète (*DWT: Discrete Wavelet Transform*) sur l'image capturée. Par la suite, les coefficients d'ondelettes sont codés en appliquant le codage SPIHT qui exploite bien la relation parent-enfant qui existe entre ces coefficients (figure 1.6).

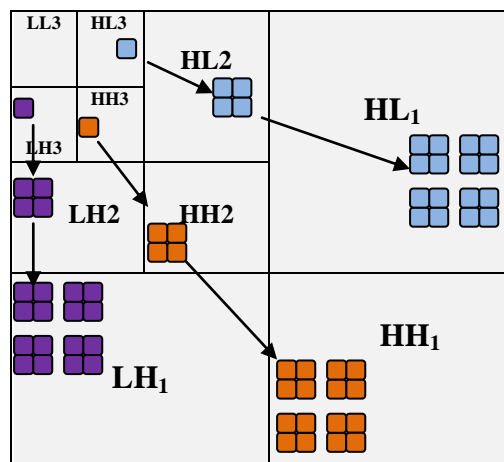


Figure 1.6 : relation parent-enfant entre les différents coefficients d'ondelettes

Un processeur *Intel StrongARM SA1110* et un émetteur radio *LMX3162* ont été utilisés pour valider l'approche utilisée. Des économies significatives d'énergie ont été réalisées, mais ces économies pourraient être améliorées si les auteurs avaient adopté une approche Hardware.

Cristian Duran-Faundez et al ont introduit dans [7] une nouvelle approche de compression à la source avec un faible coût énergétique (figure 1.7). La technique consiste tout d'abord à appliquer la transformée en ondelettes discrète sur l'image capturée puis préparer à la source des paquets de différentes priorités. Un nœud de transit qui reçoit un paquet provenant de la source décide d'acheminer ou non les données reçues en fonction de la priorité du paquet ainsi que l'état de charge du nœud. L'inconvénient de cette approche est la nécessité d'un post-traitement qui permettrait de compresser l'image efficacement. En effet, la transformée en ondelettes discrète ne représente qu'un premier étage d'une chaîne de compression d'image.

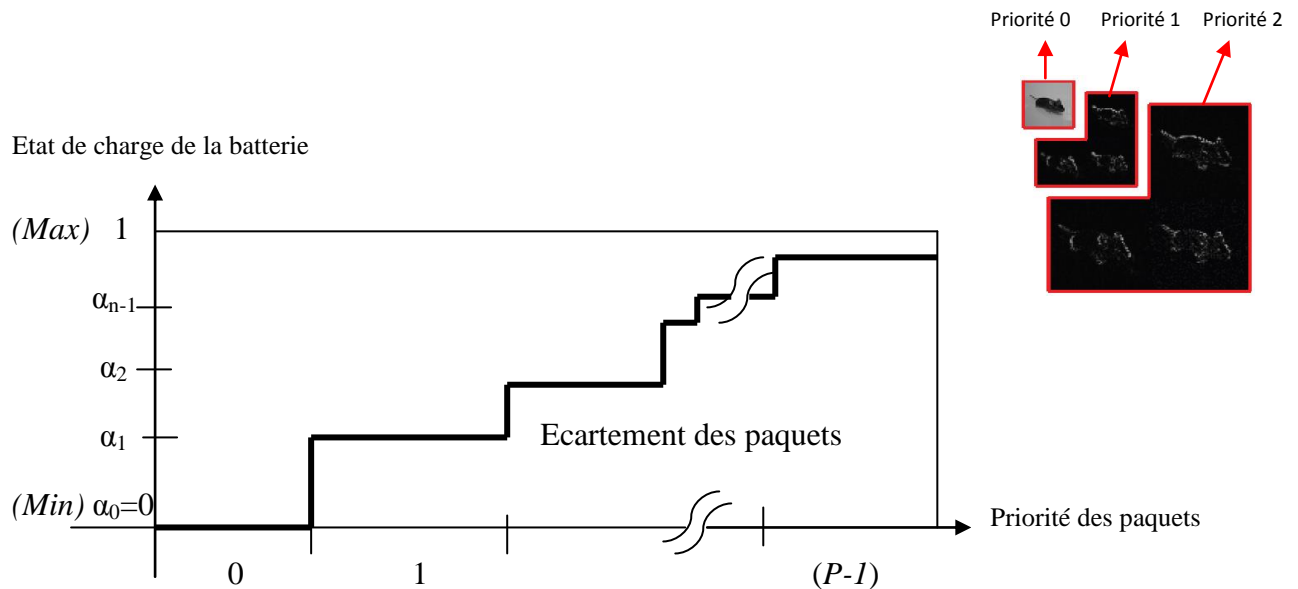


Figure 1.7 : Politique d'écartement des paquets selon l'ordre de priorité [7]

1.3.2.2. Compression distribuée d'images

La nature distribuée des réseaux de capteurs sans fil a poussé plusieurs groupes de recherche à mener des travaux qui concernent la compression distribuée d'images.

Wu et Abouzeid ont proposé de répartir le traitement lié au standard JPEG 2000 sur plusieurs nœuds du réseau [40]. Deux techniques ont été introduites :

- La première technique consiste à appliquer une transformée en ondelettes parallèle. L'image capturée est divisée en plusieurs lignes ; chaque ligne est transférée à un nœud voisin afin d'appliquer la transformée en ondelettes unidimensionnelle (1-D). Les nœuds concernés par le traitement transmettent par la suite les résultats obtenus à un nœud spécifique (appelé nœud agrégat) qui divise les nouvelles données, cette fois-ci en plusieurs colonnes. Les nouveaux blocs sont envoyés à des nœuds voisins (les mêmes nœuds ou bien d'autres nœuds qui n'avaient pas participé au traitement) pour refaire les mêmes calculs. Le nœud agrégat récupère les nouveaux résultats et nous obtenons ainsi une transformation bidimensionnelle (2-D) de l'image originale.
- La deuxième technique, appelée méthode de carrelage, consiste à diviser l'image originale en plusieurs tuiles qui seront envoyées par la suite à plusieurs nœuds du réseau. Chaque nœud doit effectuer la transformation en ondelettes 2-D indépendamment des autres nœuds du réseau. Tous les résultats seront finalement collectés par le nœud agrégat [7, 40].

Slepian et Wolf ont introduit la notion de *compression distribuée de données corrélées* [41]. La méthode proposée, pouvant être appliquée sur des données multimédias, repose sur le fait d'exploiter la corrélation existante entre plusieurs données (images par exemple) provenant de sources voisines (caméras par exemple) détectant le même processus (capture d'une scène vidéo par exemple) à partir de différents endroits (figure 1.8). Une corrélation peut aussi exister entre les données acquises successivement par un même capteur (le cas des images successives captées dans une scène vidéo). Wagner et al ont essayé de valider cette approche pour le cas particulier d'un réseau de capteurs de vision [42].

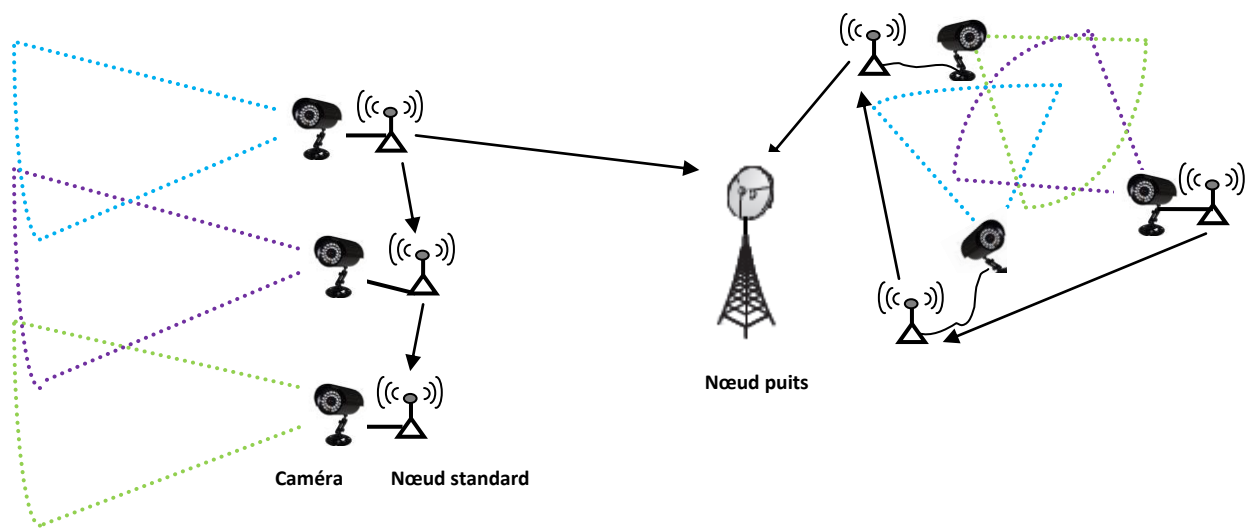


Figure 1.8 : Exemple de plusieurs capteurs de vision capturant des images corrélées

Un objet peut être capturé par plusieurs caméras voisines (deux ou plus). Au lieu d'envoyer les mêmes informations au puits, les nœuds voisins communiquent entre eux afin de déterminer les zones d'intersection. La détection du chevauchement entre les zones de couverture des différentes caméras permettra d'éviter la transmission multiple de données redondantes. Un algorithme spécifique est appliqué par le puits ou par le terminal de traitement afin de reconstruire les images provenant de sources différentes.

Qin Lu et al ont présenté dans [43] un schéma de compression distribuée basée sur la *LBT (Lapped Biorthogonal Transform)*, un algorithme de type *zero-tree* (qui exploite la relation parent-enfant entre les différents coefficients) et le codage de Golomb. La *LBT* nécessite beaucoup moins de calculs et d'espace mémoire que la *DWT*. De même, le codage de Golomb est moins complexe que le codage de *Huffman* ainsi que toutes les variantes du codage arithmétique (binaire, statique ou adaptatif). Le nœud tête de cluster (*Cluster-Head*) choisit 8 nœuds de son cluster (ayant suffisamment d'énergie) pour faire les traitements nécessaires sur l'image capturée. La caméra, dès la réception d'une requête du *Cluster-Head*,

envoie 8 lignes de données aux nœuds sélectionnés pour commencer la compression. Les nœuds effectuent la compression indépendamment les uns des autres et envoient les résultats vers le nœud tête de cluster.

Toutes ses approches de traitement distribué (compression distribué de données) n'ont pas été validées expérimentalement sur des plateformes réelles de réseaux de capteurs sans fil. Les résultats théoriques présentés dans ses références sont bonnes, mais les modèles étudiés devraient être confrontés à des résultats expérimentaux satisfaisants.

1.3.2.3. Capteurs d'image conçus spécifiquement pour les RCSFs

Peu de travaux ont porté sur la conception de capteurs d'images dédiés pour les réseaux de capteurs sans fil. Ces contributions seront détaillées dans cette partie.

Zhi-Yan Cao et Zheng-Zhou Ji ont proposé dans [44] une architecture matérielle d'un nœud de RCSF intégrant un capteur d'image CMOS, une mémoire Flash de grande taille, un circuit d'acquisition d'images, une SRAM pour accélérer les traitements effectués sur l'image capturée, un circuit de compression d'images, un processeur et un émetteur / récepteur radio à consommation réduite (Chipcon CC1000 est favorisé par rapport à d'autres émetteurs / récepteurs par le fait de consommer un courant très faible de l'ordre de 0.2 μ A en mode veille) [24]. Le schéma bloc de l'architecture proposée est présenté ci-dessous:

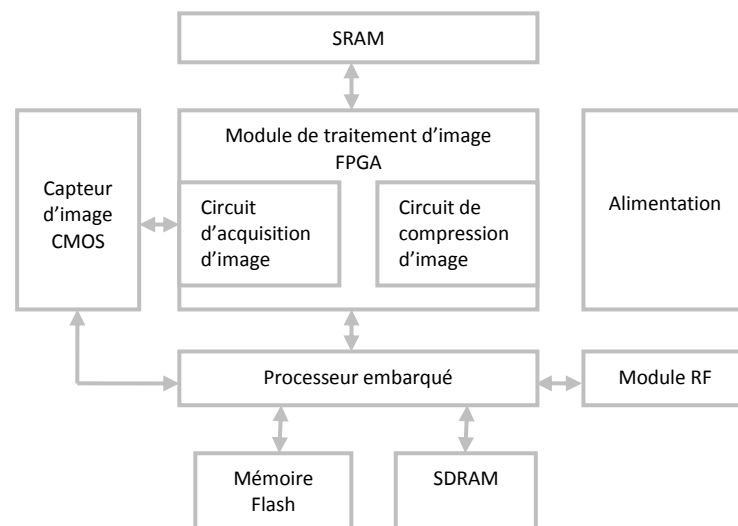


Figure 1.9 : Architecture du capteur de vision proposé [44]

L'architecture proposée nécessite une implémentation sur ASIC pour la valider. En plus, le comportement du capteur de vision conçu au sein d'un réseau de capteurs devrait être testé.

Culurciello et Andreou ont présenté dans [45] le capteur d'image ALOHA, un imageur conçu et réalisé en technologie CMOS standard 0.6µm. Le prototype, conçu spécifiquement pour les réseaux de capteurs sans fil, fonctionne comme suit : chaque pixel qui atteint un seuil de tension bien déterminé (proportionnel à la quantité de lumière collectée par la surface photosensible) envoie une requête à un circuit spécifique pour lui permettre de transmettre sa valeur. Pour résoudre le problème d'accès multiples au bus de données, la technique d'accès ALOHA est utilisée (lorsqu'un pixel a de l'information à transmettre, il envoie ses données sans se préoccuper des autres pixels. S'il y a collision, tous les pixels intervenants doivent retransmettre leurs valeurs ultérieurement). Deux versions du capteur de vision ont été développées : ALOHAim1 et ALOHAim2. La première version est un capteur d'image disposant d'une matrice de 32x32 pixels et d'un circuit détecteur de contention (ou collision) analogique. La deuxième version dispose d'une matrice de 64x64 pixels (4 quadrants de 32x32 pixels) et d'un détecteur de collision numérique. Les deux imageurs fabriqués ne permettent pas de réduire la consommation énergétique du réseau. En effet, un nombre important de paquets transmis implique une consommation élevée des nœuds du réseau. L'absence de l'aspect compression de données est l'inconvénient majeur de ce travail de recherche.

Leonardo Gasparini et al ont conçu dans [46] un nœud spécifique pour les réseaux de capteurs de vision. L'idée était d'optimiser la consommation énergétique à tous les niveaux (capture de l'image, traitement, Emetteur/Récepteur radio, etc.). Le nœud intègre un capteur d'images CMOS à faible consommation d'énergie, une unité de traitement, une unité de contrôle, une mémoire et un transcepteur radio (Emetteur/Récepteur radio). L'unité de contrôle a pour rôle d'activer l'unité de traitement et le transcepteur radio en cas de besoin (détection de mouvement dans une scène vidéo pour une application de vidéosurveillance). Ces deux unités entrent en mode veille et donc en mode faible consommation d'énergie dans les cas où il n'y a pas des données à traiter et transmettre (l'absence d'un changement significatif entre deux trames successives). L'approche a été validée en utilisant une carte FPGA (unité de contrôle, unité de traitement et la mémoire) et un capteur d'image CMOS. La durée de vie estimée du nœud pour une application de vidéosurveillance est aux alentours de 4 mois. L'absence d'une implémentation sur ASIC est l'inconvénient majeur de ce travail. En plus, les auteurs n'ont présenté aucun détail concernant la mise en place du capteur conçu dans un vrai contexte de réseaux de capteurs sans fil.

Les travaux détaillés dans cette partie présentent une nouvelle philosophie dans le domaine des réseaux de capteurs sans fil. Contrairement aux approches classiques, qui consistent à capturer puis transmettre l'image pour un post-traitement à l'extérieur, les nouvelles approches proposent qu'au moins une partie du traitement soit intégrée au sein du capteur lui-même. D'autres travaux figurant dans la littérature proposent aussi des systèmes de vision à faible consommation d'énergie et intégrant des traitements figés au sein de l'imageur. Ces travaux ont été introduits dans un contexte différent de celui des RCSFs, mais les caractéristiques attirantes de ces systèmes nous ont encouragées à exploiter cette piste. Un aperçu sur l'état de l'art des imageurs intégrant du traitement sur le plan focal sera présenté dans le chapitre suivant. L'intérêt de ce type d'imageurs pour le cas particulier des RCSFs sera aussi détaillé dans ce chapitre.

1.4. Conclusion

Le besoin de réseaux de capteurs de vision est nécessaire pour certaines applications potentielles dans un large éventail de secteurs. En effet, certaines applications potentielles (surveillance des parcs naturels, surveillance des centraux nucléaires, détection des feux de forêts, etc.) ont poussé les chercheurs à traiter le cas particulier des applications multimédias dédiées aux réseaux de capteurs sans fil. Les contraintes strictes des réseaux de capteurs rendent difficile l'implémentation de ce type d'applications. En plus, la plupart des capteurs d'image commercialisés ne sont pas adaptés aux RCSFs puisqu'ils ont des consommations énergétiques élevées. La plupart des techniques standards d'optimisation d'énergie évoquées dans ce chapitre restent loin de satisfaire les besoins spécifiques des réseaux de capteurs de vision. Seules les techniques de compression d'image et de vidéos pourraient aboutir à des résultats motivants en termes de consommation énergétique et temps de calcul, si on arrivait à bien choisir les algorithmes adéquats pour les RCSFs.

La quasi-totalité des travaux de recherche dans le domaine de compression d'image pour les RCSFs utilisent des plateformes basées sur des composants commerciaux (caméras, microcontrôleurs, mémoires, émetteurs/récepteurs, etc.) généralement non optimisés pour des traitements complexes. La conception d'un capteur d'image spécifique pour les RCSFs est donc devenue une nécessité absolue. La caméra à concevoir doit consommer peu d'énergie en mode acquisition d'image. En plus, elle doit intégrer des fonctionnalités de compression d'image afin de réduire la consommation de l'émetteur/récepteur radio, l'élément le plus gourmand en énergie dans un nœud capteur. Seule la technologie CMOS permettra de

développer des caméras ayant de telles caractéristiques ; pour cette raison, notre choix s'est limité sur les capteurs d'image CMOS. Dans le chapitre suivant, une étude détaillée des capteurs d'images CMOS sera présentée. Nous allons aussi montrer l'efficacité de l'approche proposée (capteurs d'image CMOS intégrant des fonctionnalités de compression) à travers la présentation des travaux de recherche liés à ce type d'imageurs.

Chapitre 2

Capteurs d'image CMOS pour les systèmes fortement contraints en énergie

Introduction

Les capteurs de vision classiques sont loin de satisfaire aux besoins des systèmes ayant des ressources assez limitées (énergie, mémoire, rapidité de traitement, etc.) tel que les réseaux de capteurs de vision sans fil. Les caméras standards sont dépourvues de techniques efficaces d'optimisation des ressources ce qui n'est pas appréciable dans le cas des RCSFs, systèmes fortement contraints en ressources [47, 48, 49]. La consommation énergétique reste l'un des besoins les plus prioritaires pour ce type de réseaux. Puisque le transcepteur radio est l'élément le plus gourmand en énergie dans un nœud capteur, le transfert d'images non compressées à travers le réseau épuisera rapidement les batteries de ses nœuds [7, 50]. Les progrès technologiques dans le domaine de la micro-électronique ont permis le développement de systèmes de vision intelligents intégrant du traitement au sein de l'imageur lui-même. Certains chercheurs dans le domaine des capteurs d'image CMOS ont développés des prototypes de caméras intégrant des techniques de compression d'image. La plupart de ces travaux, que nous allons détailler tout au long de ce chapitre, ont contribué de façon significative à réduire le flux de données à transmettre à travers le RCSF. Ces travaux nous ont inspiré ; nous avons donc eu l'idée d'exploiter les caractéristiques attirantes des caméras CMOS afin d'optimiser la consommation énergétique des nœuds de réseaux de capteurs de vision.

2.1. Les systèmes de vision CMOS

Les dispositifs d'acquisition d'images standards permettent de capturer une information lumineuse et de la convertir en un signal électrique. Au sein de chaque pixel du capteur d'images CMOS, les photons sont convertis tout d'abord en charges photo générées,

puis en photo courant, et enfin lus sous forme de tension analogique. Cette tension sera par la suite amplifiée et aiguillée vers un convertisseur analogique/numérique afin d'être convertie en signal numérique exploitable par les unités de calcul externes (microcontrôleur, processeur, DSP, etc).

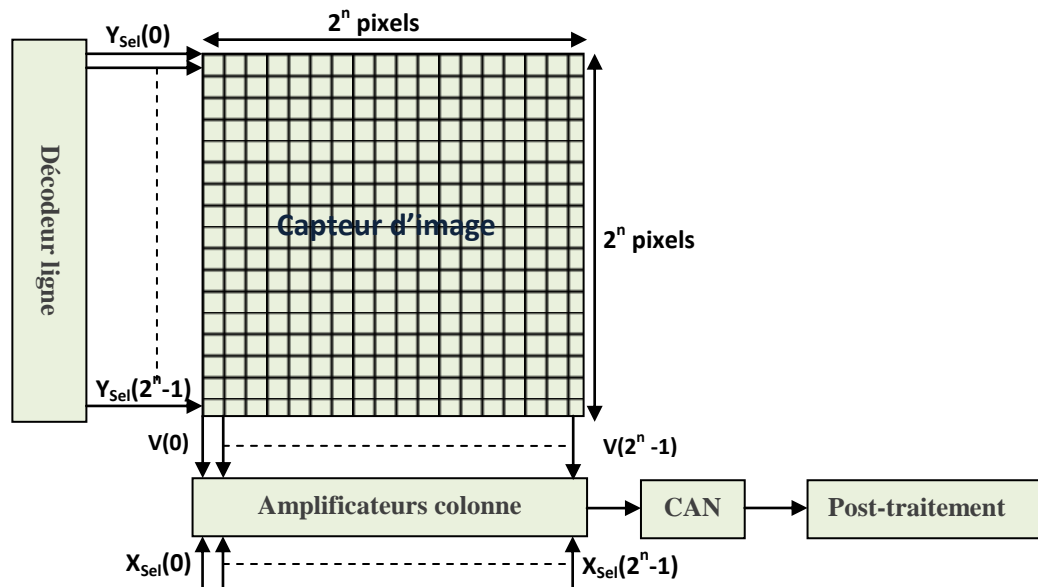


Figure 2.1 : Schéma bloc d'un système de vision CMOS classique

La figure 2.1 illustre les différents blocs d'un capteur d'image CMOS standard. Le principe de fonctionnement de chaque bloc est décrit dans cette section :

2.1.1. La matrice d'éléments photosensibles

Il s'agit de la zone photosensible du capteur. Chaque pixel comprend un photodétecteur permettant la collecte des charges électriques photo générées et quelques transistors MOS pour traiter le signal généré (conversion de la nature du signal, son amplification et sa lecture, sélection du pixel, etc.).

A. Les photodétecteurs

Le photodétecteur permet de convertir l'énergie lumineuse en énergie électrique par effet photoélectrique. Il existe trois familles de photodétecteurs : les photodiodes, les phototransistors et les photogrilles. De nos jours, la photodiode est le photodétecteur standard dans les imageurs CMOS.

L'énergie lumineuse captée est fonction du nombre de photons et la longueur d'onde de la lumière détectée (équation 2.1).

Equation 2.1.

$$E = \frac{hc}{\lambda}$$

E : énergie du photon (en joules)

h : constante de Planck ($6,623 \cdot 10^{-34}$ J.s)

c : vitesse de la lumière ($2,998 \cdot 10^8$ m.s⁻¹)

λ : longueur d'onde (m)

L'effet photoélectrique représente la création de paires *électron-trou* dans un matériau (métallique ou semi-conducteur) exposé à la lumière. La largeur de la bande interdite du matériau doit correspondre à la gamme d'énergie des photons capturés. L'énergie des photons incidents est transmise aux électrons des atomes du matériau permettant ainsi leur passage dans la bande de conduction. Sous l'effet d'un champ électrique, un courant photoélectrique proportionnel à la longueur d'onde de la lumière incidente, le flux des photons incidents et la nature du matériau sera créée [51]. Les photodétecteurs les plus utilisés dans les systèmes de vision CMOS sont la photodiode et le phototransistor, détaillés ci-dessous :

- La photodiode

En mode photoconducteur, la photodiode formée par une jonction P-N, est polarisée inverse. Un courant inverse proportionnel à l'éclairement est créé. Même en absence d'éclairement, un courant faible traversant la jonction est produit : il s'agit du courant d'obscurité, courant de nature thermo-ionique (Figure 2.2) [51].

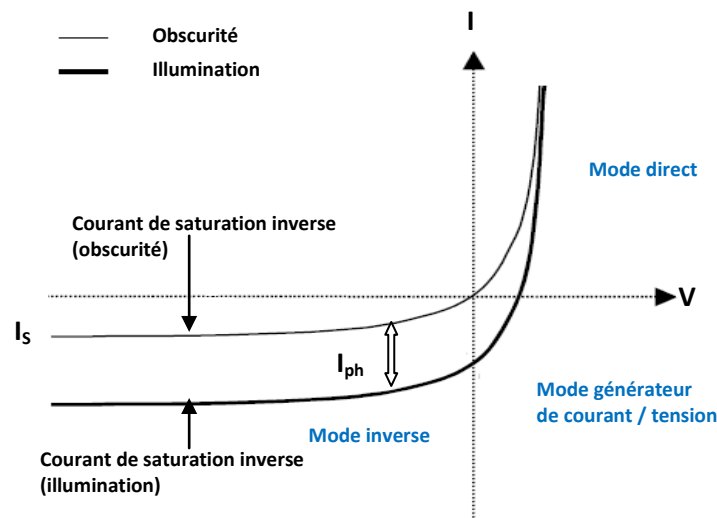


Figure 2.2 : Caractéristiques électriques de la photodiode

Equation 2.2.

$$I_{\text{obscurité}} = I_S \cdot (e^{qV/kT} - 1)$$

Equation 2.3.

$$I = I_{ph} + I_{\text{obscurité}} = I_S \cdot (e^{qV/kT} - 1) + q \cdot \eta \cdot A_d \cdot \Phi$$

I : courant de la photodiode sous illumination (A)

$I_{\text{obscurité}}$: courant d'obscurité de la diode (A)

V : tension aux bornes de la diode (V)

I_S : courant de saturation inverse de la diode (A)

I_{ph} : courant photonique (A)

q : charge élémentaire ($1,602 \cdot 10^{-19}$ C)

k : constante de Boltzmann ($1,38 \cdot 10^{-23}$ J . K⁻¹)

T : température absolue (K)

A_d : aire de la diode (m²)

Φ : flux de la lumière incidente (s⁻¹ . m⁻²)

η : rendement quantique de la photodiode

Dans le cas des imageurs CMOS, c'est le fonctionnement en mode inverse qui nous intéresse. Dans ce mode, le courant est très peu fonction de la tension aux bornes de la diode. La valeur du courant (du fA au nA) est directement proportionnelle à l'intensité de la lumière incidente [52].

- Les phototransistors

Il s'agit d'un transistor bipolaire dont le courant de base est le photo-courant généré par la jonction base-émetteur qui est polarisé en inverse. La consommation du phototransistor est supérieure à celle d'une photodiode d'un facteur β (gain du transistor). Ce photodétecteur est lent à cause de sa capacité d'entrée de grande valeur (Figure 2.3) [53].

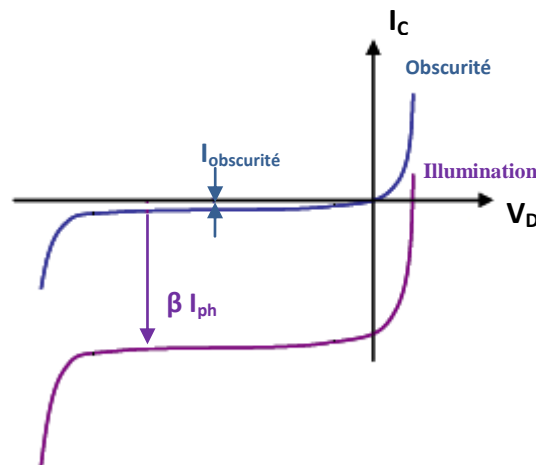


Figure 2.3 : Caractéristiques électriques d'un phototransistor

Equation 2.4.

$$I_C = \beta I_{ph} + I_{\text{obscurité}}$$

I_C : courant collecteur du phototransistor (A)
 $I_{\text{obscurité}}$: courant d'obscurité du phototransistor (A)
 I_{ph} : courant photogénéré par la jonction Base-Emetteur (A)
 V_D : tension Collecteur-Emetteur du phototransistor (V)
 β : gain du transistor

Les phototransistors ont plusieurs inconvénients (fort courant d'obscurité, mauvaise linéarité, thermo-sensibilité, etc.) ce qui rend difficile leur utilisation [51].

B. Les pixels

Deux principaux types de capteurs d'image CMOS existent (Figure 2.4): capteurs à pixels passifs (PPS : Passive Pixel Sensor) et capteurs à pixels actifs (APS : Active Pixel Sensor) :

- Pixels passifs : la lecture de l'information lumineuse se fait de manière passive en lisant tout simplement une valeur de photocourant via un transistor de sélection. Cette structure a l'avantage d'avoir un très bon taux de remplissage (un taux proche de 100%. Le taux de remplissage est défini par le rapport entre la surface photosensible du photo-détecteur et la surface du pixel), mais elle est très sensible au bruit de colonne (résultant principalement des non-uniformités des convertisseurs courant-tension) [54].
- Pixels actifs : ici la lecture est active. Le photocourant est d'abord intégré afin de créer une différence de potentiel proportionnelle à l'éclairement. Plusieurs transistors permettent par la suite de bien préparer et lire cette information. Les APS (Active Pixel Sensor) sont caractérisés par la présence d'un amplificateur suiveur au sein de chaque pixel. Cet amplificateur a pour rôle de transmettre le signal du photodétecteur sous une faible impédance au bus vertical. La conversion de l'information lumineuse en tension se fait au sein du pixel [54].

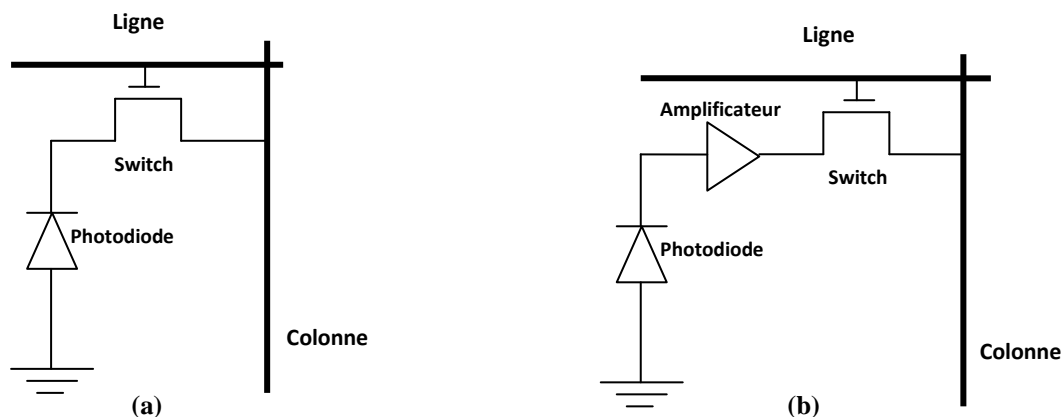


Figure 2.4 : Différents types de capteurs d'image CMOS : à pixels passifs (a) et à pixels actifs (b)

Plusieurs architectures de pixels actifs existent dans la littérature. Parmi ces architectures, on cite le 3-T APS (pixel actif à 3 transistors) à intégration et le 3-T APS à fonctionnement continu.

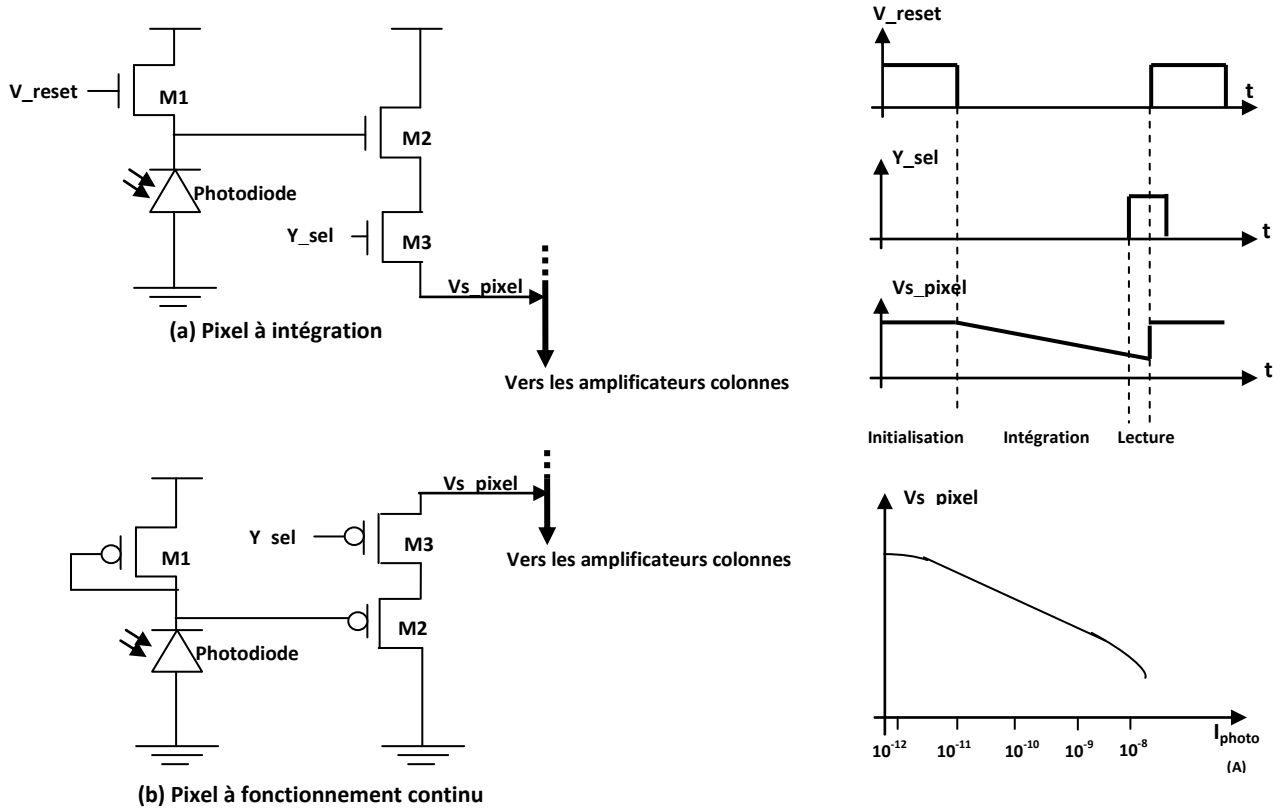


Figure 2.5 : Pixel à intégration (a) et pixel à fonctionnement continu (b)

Les pixels à intégration (figure 2.5 (a)) sont les plus courants. Ils sont industrialisés à grande échelle (appareils photo numérique, webcams, téléphonie mobile, etc.). Ce type de pixels, dans sa structure la plus simple, inclut un photo-site (la photodiode est le photo-détecteur le plus utilisé) et trois transistors. Le transistor de pré-charge ou de reset M1 polarise en inverse la photodiode formant ainsi une zone de charge d'espace où les photo-porteurs donnent naissance à un courant photo-généré. Pendant la phase d'initialisation, la photodiode est pré-chargée à la tension V_{dd} . Lorsque cette phase de reset est terminée, la phase d'intégration est activée et la décharge de la photodiode se fait proportionnellement à l'intensité lumineuse. Le transistor M2 permet l'amplification du signal de sortie du pixel alors que le transistor M3 permet la sélection du pixel.

Les pixels à fonctionnement continu ou encore pixels à compression logarithmique (figure 2.5 (b)) comportent aussi un photo-détecteur et trois transistors ; mais ils se distinguent des pixels standards (à intégration) par leur mode de fonctionnement. En effet, ils fonctionnent de façon continue (absence des phases d'initialisation et d'intégration) et la

relation qui lie la sortie du pixel avec le courant photonique suit une relation logarithmique (équation 2.5) [45, 47].

$$\text{Equation 2.5.} \quad V_{ph} = V_{dd} - n \cdot U_T \cdot \ln(I_{ph}/I_0)$$

avec $U_T = [(k.T)/q] \approx 26 \text{ mV}$ à température ambiante

V_{dd} : tension d'alimentation (V)

V_{ph} : tension aux bornes de la photodiode (V)

I_{ph} : courant photonique (A)

I_0 : courant de canal résiduel qui dépend des paramètres du procédé de fabrication (A)

n : facteur de l'effet du substrat et il varie entre 1 et 2 en fonction des paramètres du procédé de fabrication

q : charge élémentaire ($1,602 \cdot 10^{-19} \text{ C}$)

k : constante de Boltzmann ($1,38 \cdot 10^{-23} \text{ J} \cdot \text{K}^{-1}$)

T : température absolue (K)

Il existe dans la littérature d'autres architectures de pixels analogiques (APS à 4 transistors, APS à 5 transistors, etc.), mais le principe de fonctionnement est toujours le même. Le but d'ajouter un ou deux transistors supplémentaire est de réduire le bruit de lecture de l'information lumineuse ou d'améliorer la capture de scène en mouvement (Global Shutter).

Ces dernières années, on entend beaucoup parler du pixel numérique (DPS : Digital Pixel Sensor) permettant de concevoir des imageurs rapides et de grandes tailles. Dans cette nouvelle architecture (Figure 2.6), l'information lue au niveau du pixel est numérique grâce à l'intégration d'un convertisseur Analogique/Numérique élémentaire au sein de chaque pixel (parfois partagé par quelques pixels voisins). Le DPS peut intégrer également une mémoire et un processeur élémentaire. L'inconvénient majeur du pixel numérique est sa grande taille [55].

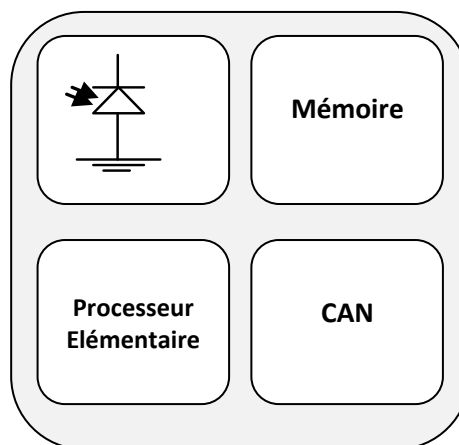


Figure 2.6 : Architecture du pixel numérique (DPS)

2.1.2. Amplificateur colonne

L'amplificateur colonne, appelée aussi circuit de lecture colonne, permet de recueillir, stocker et amplifier l'information issue du pixel. Dans un capteur d'image standard, il y a un amplificateur colonne par colonne de pixels (ce qui justifie son appellation). Tous les amplificateurs reçoivent en même temps les signaux provenant de la ligne active de l'imageur. Ces signaux seront échantillonnés puis stockés par l'ensemble des circuits de lecture colonne. Par la suite, un parcours séquentiel des amplificateurs colonne permettra de fournir les signaux amplifiés au convertisseur Analogique / Numérique.

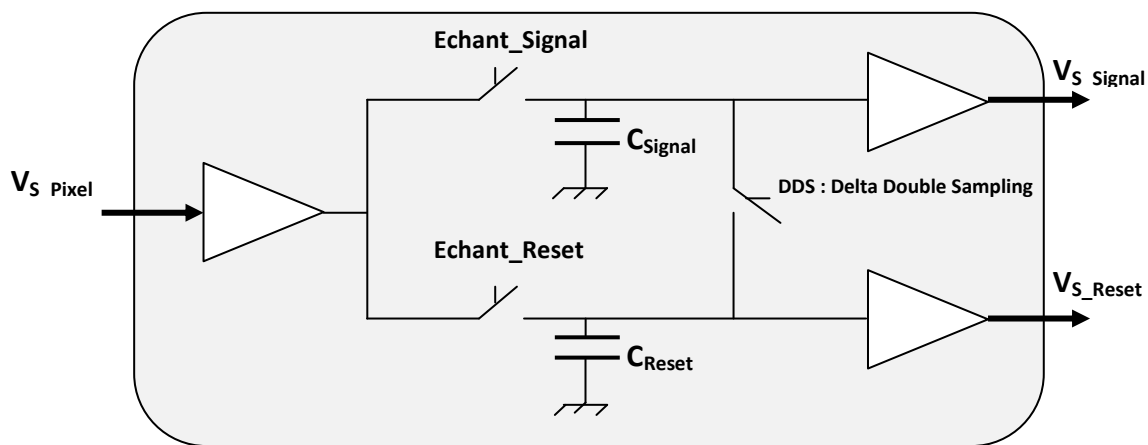


Figure 2.7 : Structure d'un amplificateur colonne standard

Le circuit de la figure 2.7 permet d'échantillonner, stocker et amplifier à la fois l'information liée à l'intensité lumineuse (C_{Signal}) et celle liée à l'initialisation (C_{Reset}). L'interrupteur DDS (la technique DDS ou Double Delta Sampling sera détaillée par la suite) sert à corriger le décalage en tension des circuits de lecture colonne. [51, 56].

2.1.3. Convertisseurs Analogiques Numériques (CAN) dédiés aux imageurs CMOS

Dans la pratique, il existe trois façons pour implémenter un CAN :

- Implémenter un seul CAN placé en sortie de l'imageur. Dans ce cas, le convertisseur devrait être très rapide afin de suivre la cadence du capteur d'image.
- Un CAN est placé au niveau de chaque colonne. La taille du convertisseur ne doit pas dépasser le pas spatial de l'imageur. La vitesse du convertisseur peut être N fois moins rapide que celle du CAN de la configuration précédente (un seul CAN pour tout l'imageur), avec N le nombre de colonnes (nombre de pixels par ligne).

- Implémenter un CAN au sein de chaque pixel (figure 2.6). La taille du convertisseur devrait être la plus petite possible ; donc on doit opter pour une architecture simple afin d'optimiser la taille du pixel. Cette architecture n'est utilisée que pour des applications d'imagerie rapide n'ayant pas de contraintes de surface et/ou de consommation [55].

Type du convertisseur	Précision	Emplacement typique
Flash	10-16 bits	Matrice
Pipeline	10-14 bits	Colonne ou matrice
Sigma-Delta	8-12 bits	Pixel ou colonne
Simple rampe	8 bits	Pixel ou colonne

Tableau 2.1 : Récapitulatif des différentes implémentations de CAN pour les capteurs d'image [55]

Le tableau 2.1 présente les types de CAN les plus utilisés par les concepteurs de capteurs d'image CMOS. Le convertisseur flash (figure 2.8) est le convertisseur le plus rapide. Il permet de convertir un signal analogique en un seul coup d'horloge grâce à un réseau de comparateurs ($2^n - 1$ comparateurs, avec n la résolution du convertisseur). Les inconvénients majeurs de ce type de convertisseurs sont la consommation énergétique élevée et l'encombrement.

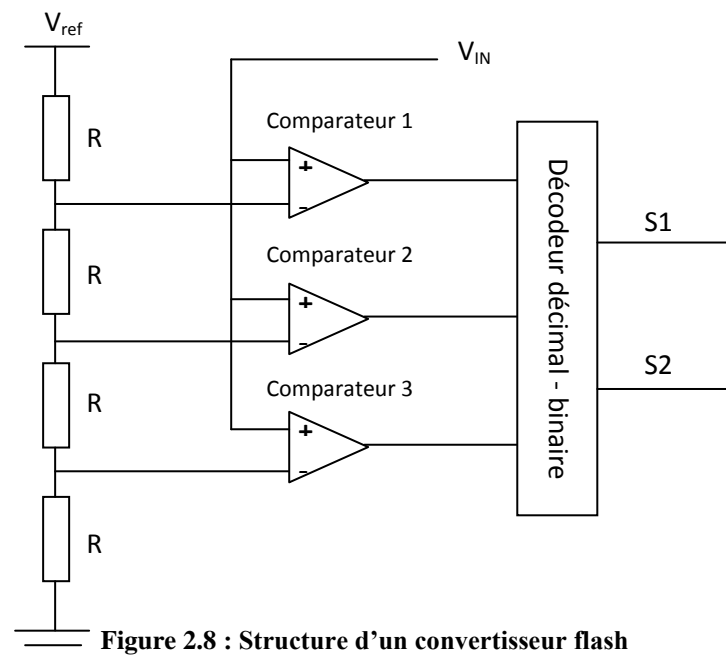


Figure 2.8 : Structure d'un convertisseur flash

Le convertisseur pipeline est aussi classé parmi les convertisseurs A/N rapides (moins rapide que le CAN de type flash mais à moindre encombrement). La structure la plus utilisée est celle du convertisseur pipeline N étages 1-bit (figure 2.9).

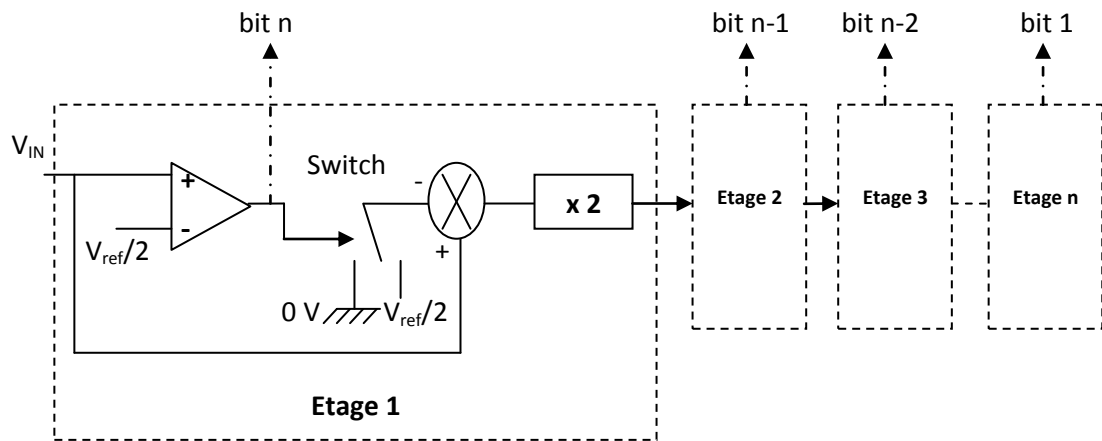


Figure 2.9 : Structure d'un convertisseur pipeline N étages 1 bit

Le convertisseur pipeline N étages 1-bit intéresse les concepteurs grâce à sa simplicité et son efficacité. Son principe de fonctionnement est très simple : le premier étage donne le bit de poids le plus fort en comparant la valeur à convertir avec la tension de référence divisée par deux ; le résultat de la comparaison permet d'aiguiller un interrupteur (Switch CMOS) qui connecte l'entrée négative du soustracteur soit à 0 V (valeur à convertir inférieure à $V_{ref}/2$) soit à $V_{ref}/2$ (valeur à convertir supérieure à $V_{ref}/2$). On retranche du signal à convertir $V_{ref}/2$ uniquement lorsque sa valeur est supérieure à $V_{ref}/2$; la nouvelle valeur obtenue sera multipliée par deux et injectée dans l'étage suivant. Chaque étage réalise le même traitement dès qu'il reçoit un signal de l'étage précédent.

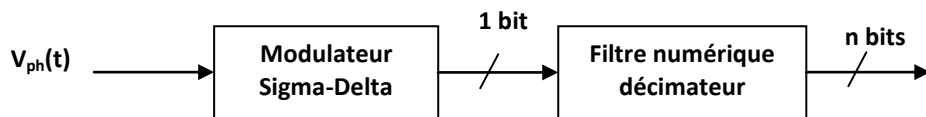


Figure 2.10 : Synoptique d'un convertisseur Analogique/Numérique Sigma-Delta

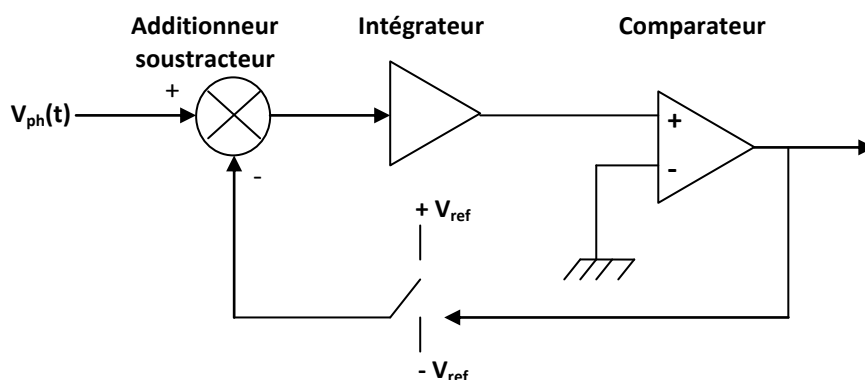


Figure 2.11 : Synoptique d'un modulateur Sigma-Delta du 1^{er} ordre

Le convertisseur Sigma-Delta (figure 2.10) est basé sur le principe de sur-échantillonnage du signal d'entrée. Un modulateur de premier ordre (figure 2.11) est généralement utilisé grâce à sa simplicité. Ce dernier comprend un additionneur soustracteur, un intégrateur, un comparateur et un convertisseur Numérique/Analogique 1-bit (le

commutateur). La sortie du modulateur fournit un flot binaire représentant $V_{ph}(t)$ sur-échantillonné à une fréquence $K.F_e$. Le comparateur convertit sur 1-bit la différence entre le signal d'entrée et le résultat de la conversion. Le commutateur prend la valeur $+V_{ref}$ si le résultat de la comparaison est "1" (le signal d'entrée est plus grand que le résultat de la conversion) et $-V_{ref}$ si le résultat de la comparaison est "0" (le signal d'entrée est plus petit que le résultat de la conversion). Le convertisseur Numérique/Analogique a donc pour rôle de faire abaisser ou augmenter la tension de sortie de l'intégrateur afin de trouver une valeur précise du signal à convertir. Donc, plus la valeur du signal à convertir (V_{ph}) est élevée, plus le nombre de "1" en sortie du modulateur soit élevé. Le filtre décimateur permettra par la suite de filtrer la sortie du comparateur en réduisant la fréquence de sur-échantillonnage à la fréquence F_e .

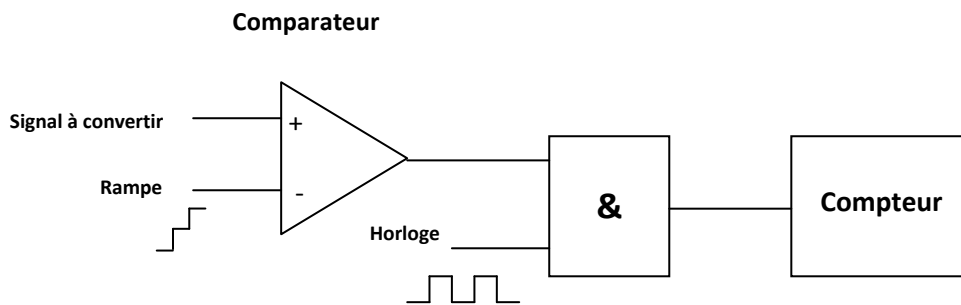


Figure 2.12: Synoptique d'un convertisseur Analogique/Numérique simple rampe

Dans le cas d'un CAN simple rampe (figure 2.12), le signal à convertir (échantillonné et bloqué par un échantillonneur-bloqueur) est comparé à un signal de type *rampe*. La sortie du comparateur et un signal d'horloge de période T permettent d'incrémenter un compteur à chaque coup d'horloge jusqu'à l'instant où la rampe atteint la valeur du signal à convertir. Malgré la simplicité de cette technique de conversion, elle présente des inconvénients multiples (lenteur, variation du temps de conversion, nécessité d'une horloge stable, et nécessité d'un échantillonneur-bloqueur rapide).

D'autres types de convertisseurs Analogiques/Numériques existent dans la littérature, mais ils sont peu exploités par les concepteurs de capteurs d'image CMOS pour des raisons multiples (coût, complexité, précision, consommation énergétique, etc.) [57].

2.1.4. Décodeurs d'adresse

Les *décodeurs d'adresse* sont les *décodeurs ligne* et les *décodeurs colonne*. Ils permettent de sélectionner l'adresse d'un seul pixel à la fois en activant les signaux Y_{sel} et

V_{reset} au niveau des pixels (figures 2.1 et 2.5) et les signaux X_{sel} , $Echant_Signal$ et $Echant_Reset$ au niveau des amplificateurs colonne (figures 2.1 et 2.7).

2.1.5. Caractéristiques générales des imageurs CMOS

Dans cette partie, nous allons détailler les principales caractéristiques qui entrent en jeu lors de la conception d'un capteur d'image CMOS.

2.1.5.1. Dynamique

La dynamique est l'aptitude d'un pixel à mesurer des éclaircissements d'une certaine amplitude. En pratique, la dynamique est définie par la relation suivante :

$$\text{Equation 2.6.} \quad \text{Dynamique}_{(dB)} = 20 \log_{10} (I_{max} / I_{min})$$

Il s'agit ici d'un rapport entre le plus grand signal non saturé détecté (le plus fort éclaircissement détecté) et le plus petit signal capturé (le plus faible éclaircissement détecté). Les capteurs d'image CMOS sont capables d'atteindre une dynamique de plus de 8 décades d'éclaircissement (capteurs d'image CMOS logarithmiques ou multi-capture) [55, 56].

2.1.5.2. Courant d'obscurité

Il s'agit du courant mesuré en absence de toute illumination. Il provient des charges thermiques circulant à travers la jonction de la photodiode. Sa valeur dépend fortement du procédé de fabrication et de la température [58].

2.1.5.3. Rendement quantique

C'est le rapport du nombre de photons pénétrant le silicium et le nombre total de photons incidents (Equation 2.7).

$$\text{Equation 2.7.} \quad \eta = (\text{Nbre d'électrons photogénérés} / \text{Nbre de photons incidents})$$

En technologie CMOS, le rendement quantique est au voisinage de 70%. Il dépend de plusieurs facteurs tels que la longueur d'onde de la lumière incidente, le matériau semi-conducteur utilisé et la concentration des porteurs libres dans le semi-conducteur [58].

2.1.5.4. Facteur de conversion

Le facteur de conversion est le gain du pixel [54]. Il s'exprime en $\mu V/e^-$ et il est défini par la relation suivante :

$$\text{Equation 2.8.} \quad \text{Fact}_{\text{conv}} = (q / C_{\text{int}})$$

q : charge élémentaire ($1,602 \cdot 10^{-19}$ C)

C_{int} : capacité d'intégration de la photodiode (f)

2.1.5.5. Vitesse de lecture

Elle s'exprime en nombre d'images par secondes et elle varie entre quelques images par seconde à plusieurs milliers d'images par seconde (10.000 images par seconde pour l'imagerie rapide). Cette vitesse dépend de plusieurs facteurs tels que le type du pixel utilisé (APS, DPS), le type (flash, simple rampe, pipeline, etc.) et le nombre de CAN utilisés (un seul CAN pour toute la matrice, 1 CAN/colonne, 1 CAN / pixel) [54].

2.1.5.6. Bruit dans les capteurs d'image CMOS

Plusieurs sources de bruit affectent les imageurs CMOS. Heureusement, les sources les plus dominantes sont dé-corrélés du signal utile. Ceci facilite la tâche de récupération de l'information utile. Dans cette section, on va présenter les différents types de bruits affectant un capteur d'image CMOS. Quelques techniques de compensation de bruit seront aussi détaillées dans cette partie [50, 53, 54, 58, 59, 60].

A. Bruit temporel

Le bruit temporel résulte de la différence de valeur entre deux échantillons successifs du même pixel sous illumination constante. Plusieurs facteurs déclenchent l'apparition du bruit temporel :

- *Le bruit thermique* : dû au mouvement aléatoire des charges sous l'effet de la température. La densité de puissance en courant du bruit thermique (A^2/Hz) est donnée par la relation suivante :

$$\text{Equation 2.9.} \quad S(f) = (4kT / R)$$

k : constante de Boltzmann ($1,38 \cdot 10^{-23}$ J . K⁻¹)

T : température absolue (K)

R : partie résistive du composant

- *Le bruit de grenaille* : il est dû à la variation du nombre de photo-porteurs dans la zone de charge d'espace sous l'effet d'un champ électrique. La densité du bruit de grenaille (A^2/Hz) est défini par la relation suivante :

$$\text{Equation 2.10.} \quad S(f) = 2qI$$

q : charge élémentaire ($1,602 \cdot 10^{-19}$ C)

I : courant qui passe à travers le composant

- *Le bruit en 1/F* : également appelé bruit rose ou bruit de scintillement, ce type de bruit est dû aux impuretés dans les composants semi-conducteurs.
- *Le bruit de génération - recombinaison* : les paires électrons-trous dissociées sous l'effet de la lumière ne génèrent pas toutes un photo-porteur capable de traverser la jonction PN. Parmi ces photo-porteurs, certains se recombinent de façon aléatoire induisant une variation du nombre de photo-porteurs traversant la jonction PN. La densité du bruit de génération - recombinaison est définie par la relation suivante :

$$\text{Equation 2.11.} \quad S(f) = K_{GR} [(4 q I_d) / (1 + (2 \pi f \tau_{GR}))]$$

q : charge élémentaire ($1,602 \cdot 10^{-19}$ C)

I_d : courant total traversant la jonction

K_{GR} : constante dépendant de la technologie utilisée et de la géométrie du composant

τ_{GR} : durée de vie des photo-porteurs, dépendant de la concentration de dopage

- *Bruit de lecture* : il est dû au pré-chargement de la capacité parasite de la photodiode. Pour faire face à ce type de bruit, il faut diminuer la valeur de la capacité parasite soit en changeant la technologie utilisée soit en modifiant la géométrie de la photodiode. La compensation du bruit de lecture par correction est aussi un moyen efficace pour restituer l'information utile.

B. Bruit Spatial Fixe (Fixed Pattern Noise)

Le bruit spatial fixe (Figure 2.13) représente la variation des valeurs des pixels sous illumination constante. Il contribue aux différences de réponses qui subsistent entre les pixels de la matrice photosensible. C'est un bruit invariant dans le temps ; il est dû aux non-uniformités des caractéristiques physiques des éléments constituant le pixel et les amplificateurs colonne (bruit de construction lié au procédé de fabrication). Deux types de FPN (*Fixed Pattern Noise*) affectent les imageurs CMOS :

- Le BSF (*Bruit Spatial Fixe*) pixel est dû d'une part aux variations des courants d'obscurité dans la photodiode et à la disparité des tailles des transistors de lecture (ce qui engendre un offset différent pour chaque pixel), et d'autre part aux variations surfaciques des photodiodes (ce qui engendre un gain différent pour chaque pixel).
- Le BSF colonne dû à la disparité des courants de polarisation des amplificateurs colonne. Ce bruit intervient de la même façon pour tous les pixels d'une même colonne (apparition de rayures verticales dans l'image).

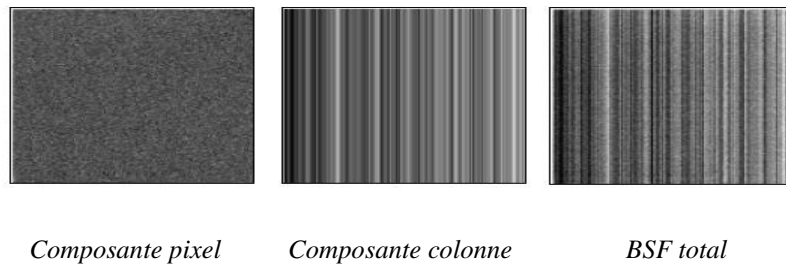


Figure 2.13: Illustration du Bruit Spatial Fixe

2.1.5.7. Compensation du bruit dans les capteurs d'image CMOS

Dans cette partie, nous allons citer les techniques de compensation de bruit utilisées lors de la conception des capteurs de vision CMOS :

A. La technique CDS (*Correlated Double Sampling*)

La technique de double échantillonnage corrélé (La technique *CDS*, Figure 2.14) permet de compenser le bruit de remise à zéro du pixel, le bruit en $1/f$ et le bruit spatial fixe des pixels. Elle consiste à effectuer deux lectures successives du même pixel pendant la lecture d'une trame. La soustraction des deux valeurs lues à deux instants différents pendant la phase d'intégration permet d'éliminer les sources de bruits indésirables. Cette technique nécessite le stockage du résultat de la 1^{ère} lecture lors de la phase d'intégration. Cet inconvénient majeur a poussé les concepteurs à abandonner cette technique et la substituer par la technique *NCDS* (*Non Correlated Double Sampling*) [61].

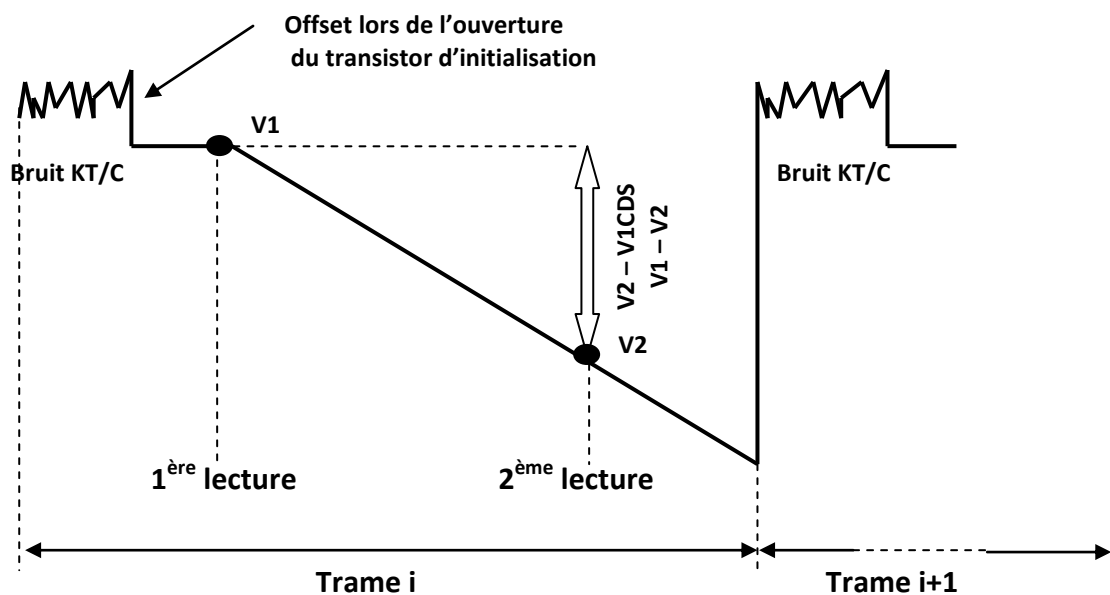


Figure 2.14: Technique de double échantillonnage corrélé (CDS)

B. La technique NCDS (Non Correlated Double Sampling)

La technique de double échantillonnage non corrélé (La technique *NCDS*, Figure 2.15) consiste à effectuer deux lectures du même pixel, mais pendant deux trames successives. La 1^{ère} lecture est faite durant la phase d'intégration de la 1^{ère} trame alors que la 2^{ème} lecture est faite durant la phase d'initialisation de la trame suivante. La soustraction des deux valeurs lues permettra par la suite de supprimer les composantes indésirables du signal. Dans le cas du *NCDS*, on n'a pas besoin de stocker la valeur de la 1^{ère} lecture. Par contre, les deux échantillons lus ne sont pas corrélés, et par conséquent leurs valeurs de bruit s'additionnent en puissance [56].

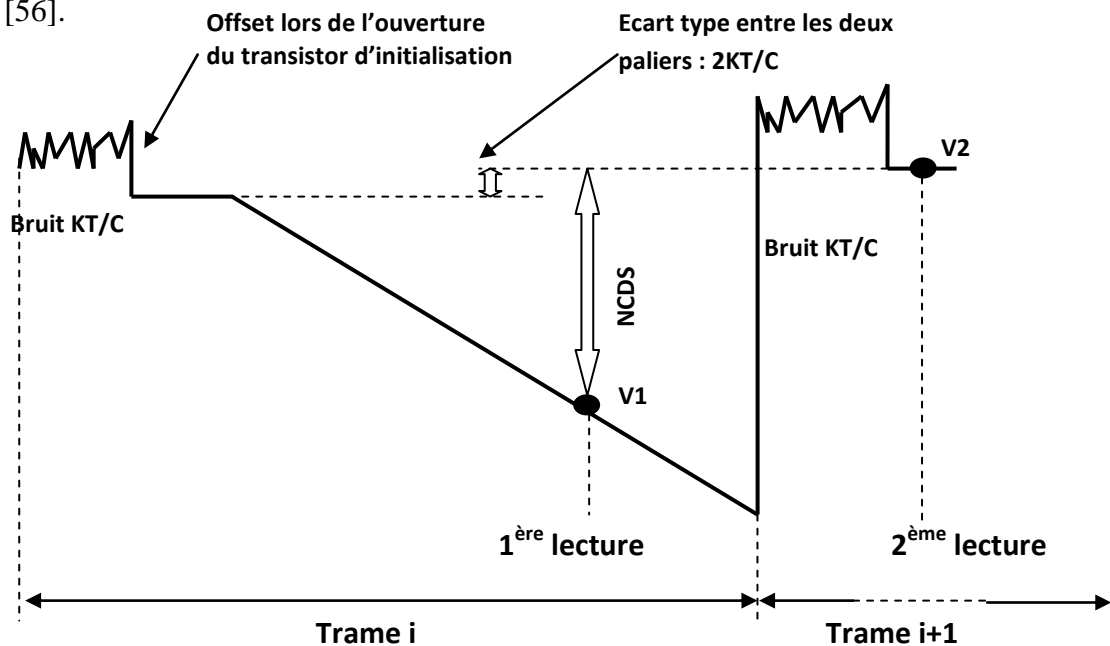


Figure 2.15: Technique de double échantillonnage non corrélé (NCDS)

C. La technique de double échantillonnage Delta (Double Delta Sampling)

La technique *DDS* (*Double Delta Sampling*) permet de compenser le bruit spatial fixe colonne (FPN colonne). La structure des amplificateurs colonne de la figure 2.7 permet d'implémenter cette technique, qui consiste à court-circuiter les entrées des amplificateurs V_{s_signal} et V_{s_reset} et mesurer par la suite le décalage en tension de leurs sorties respectives. Ceci permettra donc de corriger ce décalage en tension, mais en induisant une chute de la vitesse de lecture de la matrice de pixels (la vitesse est divisée par deux) [56].

2.2. Bilan énergétique et techniques d'optimisation pour les capteurs d'image CMOS

Tous les composants d'un système de vision CMOS contribuent de façon significative dans la consommation globale de l'imageur. L'énergie dissipée varie en fonction de l'architecture du système. La consommation peut varier de quelques microwatts pour une petite matrice de pixels actifs à 3 transistors à quelques centaines de milliwatts pour un capteur d'image numérique (DPS) large format. Une analyse théorique de la consommation énergétique des différents blocs d'un capteur de vision classique (pixel et amplificateur colonne de la figure 2.16) sera présentée dans cette partie. Quelques techniques d'optimisation de la consommation des capteurs d'image CMOS sera aussi étudiée dans cette section [63].

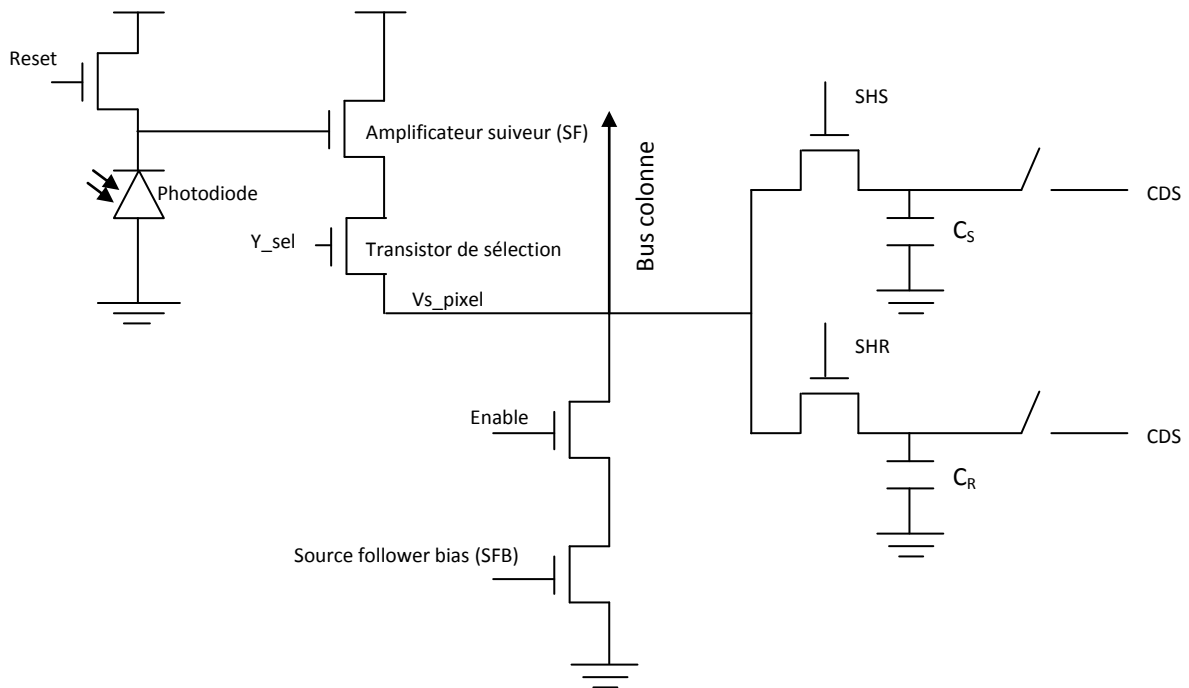


Figure 2.16 : Pixel et amplificateur colonne standards

2.2.1. Puissance dissipée par la matrice de pixels

La puissance consommée par une matrice de pixels à intégration APS-3T (*pixels actifs à 3 transistors*) utilisant le mode "capture progressive" (*rolling shutter*) est présentée par l'équation suivante :

$$\text{Equation 2.12.} \quad P_{matrice} = F_R \cdot M \cdot N (E_{Reset} + E_{SF}) + N \cdot M \cdot P_{leak}$$

F_R : nombre d'images par seconde (Frame rate)

N : nombre de lignes

M : nombre de colonnes

E_{Reset} : énergie nécessaire pour l'initialisation du pixel

E_{SF} : énergie dissipée par l'amplificateur suiveur au sein du pixel

P_{leak} : puissance dissipée due aux courants de fuite au sein du transistor de reset et du transistor de selection

$$\text{Equation 2.13.} \quad E_{Reset} = V_{dd} \cdot V_{Reset} \cdot C_{Pd}$$

V_{dd} : tension d'alimentation

V_{Reset} : tension de reset du pixel

C_{Pd} : capacité équivalente de la photodiode

Dans le cas où on utilise un transistor d'initialisation de type NMOS, la tension V_{Reset} est approximativement égale à V_{dd} . Durant la phase d'initialisation du pixel, une partie de l'énergie provenant de la source d'alimentation est stockée au sein de la capacité C_{Pd} de la photodiode, alors que l'autre partie est dissipée par le transistor de reset. Durant la phase de photo-transduction, l'énergie conservée par la capacité C_{Pd} est dissipée par le mécanisme de photo-génération.

L'amplificateur suiveur au sein de chaque pixel a pour rôle de transmettre un signal analogique sous une impédance réduite vers les amplificateurs colonne (échantillonnage, stockage et amplification du signal). L'énergie dépensée se traduit par l'équation suivante:

$$\text{Equation 2.14.}$$

$$E_{SF} = V_{dd} \cdot V_S (C_S + C_{col}) + V_{dd} \cdot V_R (C_R + C_{col}) + I_{LN} \cdot V_{DSF_Sav} \cdot T_{CS_charge} \\ + I_{LN} \cdot V_{DSF_Rav} \cdot T_{CR_charge} + I_{LN} (V_{dd} - V_S) \cdot \alpha_S + I_{LN} (V_{dd} - V_R) \cdot \alpha_R$$

C_S : capacité pour stocker le signal liée à l'intensité lumineuse

C_R : capacité pour stocker le signal liée à l'initialisation du pixel (signal de reset)

C_{col} : capacité du bus colonne

V_S : valeur maximale de la tension aux bornes de la capacité C_S

V_R : valeur maximale de la tension aux bornes de la capacité C_R

I_{LN} : courant de polarisation fourni par le transistor SFB

T_{CS_charge} : temps nécessaire pour charger la capacité C_S

T_{CR_charge} : temps nécessaire pour charger la capacité C_R

V_{DSF_Sav} : valeur moyenne de la tension drain-source du transistor SF durant le chargement de la capacité C_S

V_{DSF_Rav} : valeur moyenne de la tension drain-source du transistor SF durant le chargement de la capacité C_R

$\alpha_S = T_{S_read_out} - T_{CS_charge}$; avec $T_{S_read_out}$ est la période durant laquelle le transistor de sélection est fermé pour permettre la lecture du signal utile (proportionnel à l'intensité lumineuse)

$\alpha_R = T_{R_read_out} - T_{CR_charge}$; avec $T_{R_read_out}$ est la période durant laquelle le transistor de sélection est ouvert pour permettre la lecture du signal de reset

Dans la pratique, les concepteurs des capteurs d'image CMOS font le choix symétrique suivant :

$$\alpha_S = \alpha_R = \alpha_{S/R} ; C_S = C_R = C_{S/R} ; V_S = V_R = V_{S/R} ; V_{DSF_Sav} = V_{DSF_Rav} = V_{DSF_S/Rav} ; T_{CS_charge} = T_{CR_charge} \\ = T_{CS/R_charge}$$

L'équation 2.14 devient :

Equation 2.15.

$$E_{SF} = 2 \cdot V_{dd} \cdot V_{S/R} (C_{S/R} + C_{col}) + 2 \cdot I_{LN} \cdot V_{DSF_S/Rav} \cdot T_{CS/R_charge} + 2 \cdot I_{LN} (V_{dd} - V_{S/R}) \cdot \alpha_{S/R}$$

Les deux premiers termes de l'équation 2.15 représentent la puissance nécessaire dissipée pour la lecture du pixel. Le dernier terme devrait être optimisé afin de réduire la consommation énergétique totale de la matrice. Dans la pratique, l'optimisation significative de la consommation totale de l'imageur passe par l'optimisation du premier terme. En effet, la capacité C_{col} du bus colonne devient de plus en plus importante pour les imageurs de large format. La conception de capteurs d'image CMOS de petite et moyenne taille est appréciable puisqu'elle permet des économies importantes d'énergie.

La puissance dissipée " P_{leak} " (dernier terme de l'équation 2.12) est due aux courants de fuite à travers le transistor de reset. Ces courants de fuite deviennent de plus en plus importants avec les technologies submicroniques récentes (à partir du 0.18 μm). Il existe trois composantes principales de courants de fuite : courant de fuite de grille, courant sous-seuil et courant de fuite entre source et drain. Actuellement, les fabricants des capteurs d'image CMOS évitent de concevoir des imageurs en technologies submicroniques récentes (procédés plus récents que la technologie 0.13 μm) à cause de la dégradation significative des performances du capteur. Le "courant sous-seuil du transistor" reste prédominant par rapport aux autres courants de fuite dans les capteurs d'image CMOS actuels. Le courant sous le seuil d'un MOS est donné par l'équation suivante:

Equation 2.16.

$$I_{sub_th} = I_0 \cdot \exp[(V_{GS} - V_{TH}) / (n \cdot V_t)] \cdot (1 - \exp[-V_{DS} / V_t]) \cdot \exp[(\eta \cdot V_{DS}) / (n \cdot V_t)]$$

$$I_0 = \mu_0 \cdot C_{ox} \cdot (W/L) \cdot (n - 1) \cdot V_t^2$$

V_{TH} : tension de seuil du transistor

$V_t = kT / q$; tension thermique

η : coefficient de l'effet d'abaissement de la barrière de potentiel induite par le drain (DIBL : Drain Induced Barrier Lowering)

n : coefficient de basculement sous-seuil du transistor (subthreshold swing coefficient)

μ_0 : constante magnétique ou perméabilité magnétique du vide ($\mu_0 = 4\pi \cdot 10^{-7}$ Kg. m. A⁻². S⁻²)

C_{ox} : capacité de l'oxyde par unité de surface

I_{sub_th} varie en fonction de l'intensité lumineuse du pixel. Ceci s'explique par la dépendance de ce courant de fuite avec la tension V_{DS} du transistor de reset. La variation du

courant I_{sub_th} a un impact considérable non seulement sur la puissance dissipée au sein du pixel, mais aussi sur la qualité de l'image capturée.

En ce qui concerne les imageurs qui emploient le mode "*capture instantanée*", la puissance dissipée est similaire à celle des imageurs utilisant la "*capture progressive*". La seule différence est la présence d'un pic élevé de consommation énergétique, dû à la lecture simultanée de tous les pixels de l'imageur (équation 2.17).

$$\text{Equation 2.17.} \quad E_{Peak_Reset} = N \cdot M \cdot V_{dd} \cdot V_{Reset} \cdot C_{Pd}$$

Les capteurs d'image numériques intègrent du traitement, de la mémoire et de la conversion analogique / numérique au sein de chaque pixel. Cette catégorie d'imageurs a une consommation énergétique très élevée ; plusieurs blocs analogiques et numériques présents au sein de chaque pixel fonctionnent en même temps provoquant un fort appel de courant.

La puissance dissipée par un capteur d'image numérique est traduite par l'équation suivante :

$$\text{Equation 2.18.} \quad P_{matrice} = F_R \cdot M \cdot N (E_{Reset} + E_{read_out} + E_{analog} + E_{digital}) + N \cdot M \cdot P_{leak}$$

E_{read_out} : énergie dissipée lors de la lecture d'un pixel ($E_{read_out} = E_{SF}$ lorsqu'il s'agit d'un capteur standard)

E_{analog} : énergie dissipée par les blocs analogiques au sein de chaque pixel

$E_{digital}$: énergie dissipée par les blocs numériques au sein de chaque pixel

2.2.2. Puissance dissipée dans les décodeurs d'adresse

Les décodeurs d'adresse permettent de sélectionner une seule adresse de pixel à la fois. La puissance consommée par ce bloc est traduite par l'équation suivante :

$$\text{Equation 2.19.} \quad P_{scan} = P_t (C_L \cdot V \cdot V_{dd} \cdot f_{clk}) + I_{SC} \cdot V_{dd} + P_{leak}$$

C_L : énergie dissipée lors de la lecture d'un pixel ($E_{read_out} = E_{SF}$ lorsqu'il s'agit d'un capteur standard)

f_{clk} : fréquence de l'horloge

P_t : probabilité d'occurrence de l'événement " sélection d'une ligne " (ou d'une colonne), appelé encore facteur d'activité

V : tension de basculement, généralement égale à la tension V_{dd} en technologie CMOS

I_{SC} : courant de court-circuit. Il apparaît quand plusieurs transistors NMOS et PMOS sont actifs simultanément, induisant le passage direct du courant d'alimentation vers la masse.

Le premier terme de l'équation 2.19 est lié à la puissance dissipée lors de la phase de commutation des transistors (pour sélectionner une ligne ou une colonne). Le deuxième terme de l'équation représente la puissance due au courant de court-circuit I_{SC} . Aucun des trois termes de l'équation ne peut être négligé. En effet, la fréquence d'activité des décodeurs d'adresse n'est pas trop élevée dans le cas des capteurs d'image CMOS, ce qui fait que la

puissance dissipée lors de la phase de commutation est comparable aux autres puissances consommées (liées aux courants de fuite et de court-circuit).

Le facteur d'activité et la fréquence de fonctionnement des décodeurs d'adresse ligne et colonne sont définis par les équations suivantes :

$$P_{t_row} = 1/N ; P_{t_column} = 1/M$$

$$f_{row} = F_R \cdot N ; f_{column} = F_R \cdot N \cdot M$$

L'équation 2.19 devient donc :

$$\text{Equation 2.20. (Décodeurs ligne)} \quad P_{row_scan} = C_L \cdot V_{dd}^2 \cdot F_R + I_{SC} \cdot V_{dd} + P_{leak}$$

$$\text{Equation 2.21. (Décodeurs colonne)} \quad P_{col_scan} = C_L \cdot V_{dd}^2 \cdot F_R \cdot N + I_{SC} \cdot V_{dd} + P_{leak}$$

2.2.3. Puissance dissipée dans les circuits analogiques "Front End " (amplificateur colonne)

La puissance dissipée dépend de l'architecture des circuits *Front End*. Elle est généralement traduite par la formule généralisée suivante :

$$\text{Equation 2.22.} \quad P_{AFE} = P_{SF_bias} + P_{CDS} + P_{amp}$$

P_{SF_bias} : puissance dissipée dans le transistor SFB (situé au pied de la colonne de l'imageur)

P_{CDS} : puissance dissipée dans le circuit CDS

P_{amp} : puissance dissipée dans l'amplificateur colonne

La puissance P_{SF_bias} dépend à la fois du courant I_{LN} qui traverse le transistor *SFB* et de la tension *drain-source* du transistor. Cette différence de potentiel varie en fonction de la valeur des deux signaux V_{Reset} (tension de reset) et V_{s_pixel} (valeur proportionnelle à l'intensité lumineuse du pixel). P_{SF_bias} atteint sa valeur maximale pour une tension V_{DS} égale à $(V_{dd} - 2 V_{TH})$. Dans ce cas particulier, la puissance P_{SF_bias} peut être estimée par la valeur suivante :

$$\text{Equation 2.23.} \quad P_{SF_bias} = N \cdot 2 \cdot I_{LN} \cdot (V_{dd} - 2 V_{TH}) \cdot T_{read_out} \cdot F_R \cdot M$$

$$T_{read_out} = T_{S_read_out} = T_{R_read_out}$$

2.2.4. Puissance nécessaire pour la conversion Analogique / Numérique

La conversion Analogique / Numérique apporte une contribution élevée à la puissance totale dissipée par l'imageur. Le choix d'une architecture efficace (rapidité, précision) à faible coût (faible puissance dissipée et surface raisonnable du convertisseur) reste un enjeu pour les

concepteurs de capteurs d'image. La caractéristique la plus significative pour les concepteurs est l'énergie consommée par étape de conversion (équation 2.24).

$$\text{Equation 2.24.} \quad E_{Conv} = P_D / (2^N \cdot f_s)$$

P_D : puissance consommée par le CAN

N : résolution du CAN

f_s : fréquence d'échantillonnage

Dans la littérature, le *CAN à approximations successives* et le *CAN Delta-Sigma* sont les deux types de convertisseurs A/N les mieux réputés pour leurs caractéristiques attirantes en termes de faible consommation, résolution optimale, surface et fréquence d'échantillonnage raisonnables.

2.2.5. Techniques d'optimisation de la puissance dissipée dans les imageurs CMOS

La réduction de la consommation énergétique dans les capteurs d'image CMOS devrait être appliquée à tous les niveaux [62, 63]:

2.2.5.1. Optimisations liées aux choix technologiques

Le choix du process technologique adéquat est très important. La technologie employée doit permettre la fabrication de capteurs d'image performants et à faible coût (consommation et surface). L'implémentation d'imageurs en technologies submicroniques très avancées (plus récentes que la technologie 0.18 μm) n'est pas appréciée. Des dégradations significatives des caractéristiques de certains systèmes de vision CMOS, fabriqués en technologies très fines, ont été observées (courants de fuite élevés, faible SNR, faible photosensibilité, etc.).

Récemment, plusieurs capteurs d'image CMOS ont été implémentés en technologie *SOS* (*Silicon-On-Sapphire* ou *Silicium sur Saphir*), un des procédés de fabrication les plus maîtrisés de la technologie *SOI* (*Silicon-On-Insulator* ou *Silicium sur Isolant*). Le saphir, nommé aussi oxyde d'aluminium (Al_2O_3), est utilisé comme substrat dans la technologie *SOS*. Cette technologie est caractérisée par une très faible consommation énergétique due aux faibles valeurs des courants de fuite et des capacités parasites. Le substrat transparent des *wafers SOS* permet de concevoir des imageurs CMOS avec rétro-éclairage (éclairage par l'arrière) et par conséquent un meilleur contraste et une meilleure sensibilité à la lumière.

Quelle que soit la technologie employée, il est recommandé que le procédé de fabrication choisi puisse satisfaire certaines caractéristiques qui optimisent l'énergie dissipée :

- Possibilité d'implémenter des transistors ayant différentes tensions de seuil.
- Possibilité de faire fonctionner le circuit à faible tension.
- Possibilité d'implémenter des transistors à oxyde de grille épais afin de minimiser le courant de fuite de grille.
- Possibilité d'implémenter des capacités de faibles valeurs.
- Possibilité de séparation entre les polarisations de substrat des transistors NMOS et ceux des transistors PMOS.

2.2.5.2. Optimisations aux niveaux algorithmique et architectural

L'optimisation au niveau algorithmique consiste à réduire au maximum le nombre d'instructions sans toucher au bon fonctionnement de l'algorithme. Réduire la puissance dissipée au niveau algorithmique nécessite de suivre les consignes suivantes :

- Réduire l'activité des différents blocs de traitement.
- Détecter les régions d'intérêt et couper l'alimentation sur les circuits de lecture du reste de la matrice.
- Transférer des images entières pour un post-traitement uniquement en cas d'indispensabilité.
- Effectuer les calculs intensifs en dehors de la puce (post-traitement).

Le niveau architectural est fortement dépendant du niveau algorithmique. Par conséquent, l'optimisation au niveau architectural peut influencer et même modifier l'algorithme à implémenter. Une attention particulière doit donc être prise en compte à ce niveau. Les règles de conception suivantes peuvent réduire efficacement la puissance dissipée :

- Mettre en veille ou désactiver les blocs qui sont temporairement inactifs.
- Utiliser les registres à décalage pour la lecture des pixels.
- Renforcer les traitements parallèles au pied de la colonne de la matrice de pixels.
- Optimiser le nombre et l'emplacement des blocs du système de vision à concevoir.
- Réduire au maximum le nombre et la taille des bus.
- Rechercher la meilleure adéquation Algorithme / Architecture.

2.2.5.3. Optimisations aux niveaux circuit et logique

Plusieurs techniques d'optimisation sont utilisées afin de réduire la consommation énergétique au niveau circuit et au niveau logique. Parmi ces techniques, on peut citer :

A. Contrôle des courants de fuite

La réduction des courants de fuite peut être accomplie par l'une des actions suivantes :

- Différenciation entre le mode "actif " et le mode "veille " en insérant un transistor de mise en veille (le cas du transistor en série avec le transistor SFB de la figure 2.16).
- Utilisation des transistors empilés. Cette technique permet d'optimiser le courant de fuite de type "courant sous seuil" en réduisant simultanément les tensions V_{GS} , V_{DS} et la tension de polarisation du substrat.
- Utilisation des techniques de contrôle des courants de fuite qui ne sont pas utilisés par les concepteurs des capteurs d'image CMOS. Parmi ces techniques on cite *MTCMOS* (*Multi-threshold voltage CMOS*) et *VTCMOS* (*Variable-threshold CMOS*), techniques qui rendent possible la variation de la tension de seuil du transistor.

B. Conception de circuits opérationnels à faible tension

Un moyen efficace pour réduire la consommation énergétique est d'utiliser l'adaptation dynamique en tension d'alimentation (*Dynamic Voltage Scaling ou DVS*) pour contrôler dynamiquement l'énergie consommée. Dans un circuit CMOS, l'énergie consommée E est proportionnelle au carré de la tension V_{dd} et à la capacité totale commutée C (équation 2.25). La variation de la vitesse est donnée, quant à elle, par l'équation 2.26. Les paramètres α et β sont des constantes technologiques positives.

$$\text{Equation 2.25.} \quad E = \alpha C V_{dd}^2$$

$$\text{Equation 2.26.} \quad S = \alpha (V_{dd} - \alpha)^\beta$$

Une attention particulière doit être prise en compte lors de l'utilisation de la technique *DVS*. Cette technique ne doit pas être utilisée dans les blocs critiques du circuit afin d'éviter une dégradation significative du fonctionnement du circuit.

C. Polarisation du substrat

La technique de polarisation du substrat (*Adaptive Body Biasing ou ABB*) consiste à dissocier le substrat des transistors et leurs sources afin d'y appliquer une tension V_{bs} et réduire, à V_{dd} constante, le courant de fuite I_{DS_leak} entre le drain et la source. On diminue ainsi l'énergie statique :

$$\text{Equation 2.27.} \quad E_{stat} = \int_t V_{dd} I_{dsleak} dt$$

Le courant de fuite dépend aussi de la tension d'alimentation V_{dd} (équation 2.28). Les paramètres a , b et c sont positifs et dépendent de la technologie utilisée.

$$\text{Equation 2.28.} \quad I_{dsleak} = a.e^{b.V_{dd}}.e^{-c.V_{bs}}$$

Ainsi, à V_{dd} fixe, plus on augmente V_{bs} plus les courants de fuite diminuent. Au contraire, à V_{bs} fixe, plus on augmente V_{dd} , plus les fuites augmentent.

Certes, toutes les techniques d'optimisation de la consommation d'énergie citées ci-dessus permettent de réduire de façon considérable la consommation des capteurs d'image; mais, dans le cas particulier des réseaux de capteurs de vision sans fil, la raréfaction des flux de données sortant de l'imageur est le moyen le plus efficace pour minimiser l'énergie consommée par les nœuds de réseaux de capteurs de vision sans fil. En effet, l'émetteur/récepteur radio dans un nœud de réseau de capteurs sans fil est le bloc qui consomme le plus d'énergie; l'énergie dissipée pour transmettre un bit pourrait être utilisée pour exécuter des centaines, voire des milliers d'instructions [8]. La conception d'un imageur CMOS intégrant des fonctionnalités de compression d'image permettrait donc de réduire le nombre de paquets échangés à travers le réseau, et par conséquent prolonger sa durée de vie. Dans la section suivante, nous allons citer quelques travaux de recherche liés aux capteurs d'image CMOS exploitant la compression de données.

2.3. Capteurs d'image CMOS intelligents pour les systèmes à ressources limités

Le capteur d'image standard reste loin de répondre aux besoins spécifiques des RCSFs. Les systèmes de vision intelligents s'avèrent plus intéressants puisqu'ils permettent d'intégrer des traitements spécifiques, notamment la compression d'image, au sein de l'imageur lui-même. Afin de raréfier les données avant qu'elles soient transmises à travers un RCSF, des traitements spatio-temporels devraient être appliqués sur le flot d'images acquises. L'intégration d'algorithmes de compression efficaces pourrait donc aboutir à une optimisation de la consommation liée au transfert de l'information [64, 65].

Les recherches actuelles s'orientent vers les capteurs d'images CMOS intégrant des fonctionnalités et des traitements dédiés pour les systèmes à contraintes strictes (mémoire, autonomie de la batterie, coût, etc.). Le cas particulier des réseaux de capteurs de vision sans fil a vivement intéressé la communauté scientifique. En effet, les RCSFs ont des contraintes spécifiques (bande passante limitée, faible autonomie énergétique, mémoire de faible taille, etc.) qui rendent difficile la mise en place d'applications multimédias à travers ce type de

réseaux. Dans cette section, nous allons présenter quelques idées développées dans la littérature (capteurs de vision intelligents intégrant des traitements spécifiques), permettant de faciliter l'implémentation d'applications liées aux traitements d'images et de vidéos au sein de ces systèmes à contraintes multiples.

Plusieurs chercheurs ont proposé de raréfier les données avant de les transmettre à une unité de calcul externe à l'imageur. Malgré que la plupart de ces solutions n'aient pas été développées spécifiquement pour les réseaux de capteurs sans fil, les idées proposées peuvent servir de modèles pour les réseaux de capteurs de vision du futur. En effet, l'émetteur/récepteur radio est le bloc le plus gourmand en énergie au sein d'un nœud de RCSF ; l'intégration, au sein des nœuds du réseau, d'un imageur intelligent implémentant des fonctionnalités de compression basées sur des traitements spatiaux ou spatio-temporels pourrait faire face aux contraintes strictes de ce type de réseaux (consommation énergétique, délai de transmission, espace mémoire, etc.).

Shoushun Chen et al [66] ont conçu un capteur d'image CMOS incluant une matrice photosensible de 64x64 pixels numériques (photodiode, comparateur et RAM statique 8-bit au sein de chaque pixel) et un processeur (intégrant plusieurs blocs) permettant la compression efficace de l'image : le traitement consiste à implémenter un bloc de quantification adaptative basée sur les techniques FBAR (Fast Boundary Adaptation Rule) et DPCM (Differential Pulse Code Modulation), suivi par une transformation QTD (Quadrant Tree Decomposition). Le processeur de compression d'image implémenté a permis d'atteindre un taux de compression de 0.6 bpp et d'avoir une surface en silicium de 1.8 mm² et une puissance consommée de 6.3 mW. Par contre, le nombre de transistors par pixel est relativement élevé ce qui induit un faible facteur de remplissage (12 %).

Zhiqiang et al [67] présentent un imageur CMOS intégrant une technique efficace de compression implémentée au pied de la colonne de la matrice de pixels (Figure 2.17). La technique employée est basée sur le codage SPIHT (Set Partitioning In Hierarchical Trees), qui est une variante améliorée du codage EZW et qui s'applique généralement sur toute l'image transformée (transformée en ondelettes, QTD, etc.) ce qui nécessite une mémoire de grande taille afin de stocker à la fois le code résultant et les données temporaires liées à la compression. Pour remédier à ce problème, Zhiqiang et al proposent d'implémenter 7 processeurs SPIHT au pied de la colonne de l'imageur (un nombre de processeurs équivalent au nombre de blocs par ligne). Les processeurs fonctionnent en parallèle ; chacun d'eux traite

un bloc de 8x8 pixels. Par conséquent, la complexité liée au traitement sera réduite du fait que le nombre de descendants de chaque pixel est minimisé. En plus, le traitement parallèle accélère le traitement ce qui permet d'augmenter le nombre de trames acquises par seconde. Les résultats de simulation sous *Matlab* et sous *Cadence* étaient prometteuses : la qualité des images reconstruites est bonne (un PSNR de 52.27 dB si on applique la technique CDS et un PSNR de 50.24 dB dans le cas contraire, pour l'image de référence Lena).

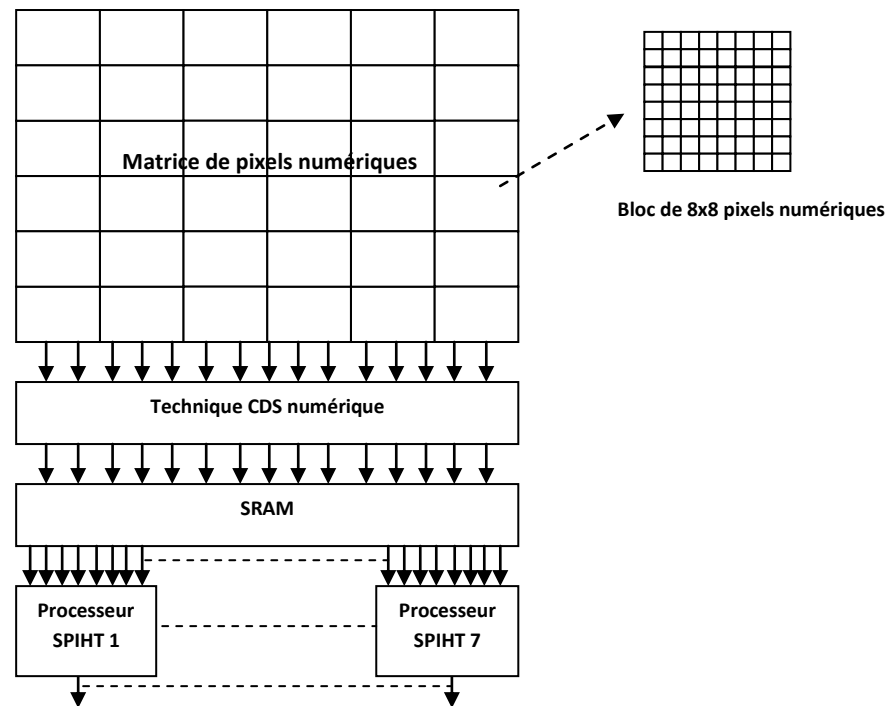


Figure 2.17 : Imageur CMOS intégrant 7 processeurs SPIHT au pied de la colonne [67]

Shoji Kawahito et al [68] ont fabriqué un capteur d'image CMOS incluant une matrice de pixels de taille 128x128 et un processeur analogique pour appliquer la DCT-2D sur les blocs de l'image acquise (Figure 2.18). Le processeur permet de traiter tous les blocs (de taille 8x8 pixels) de la matrice, un par un. A chaque fois qu'un bloc est traité, une étape de quantification à seuil variable et une étape de conversion Analogique/Numérique auront lieu au sein du CAN. La sortie du CAN sera traité par un codeur entropique de type RLC (Run Length Coding) suivi par le codage de Huffman. Les résultats obtenus lors de l'implémentation du circuit en technologie 0.35 μm sont motivants : la qualité de l'image reconstruite est bonne (un PSNR de 31.4 dB sans compensation de l'offset et 36.7 dB avec compensation), la puissance consommée par le processeur dédié à la DCT-2D est de 10.8 mW, le facteur de remplissage est de 56.6 % et le bruit spatial fixe est faible (0.38 %).

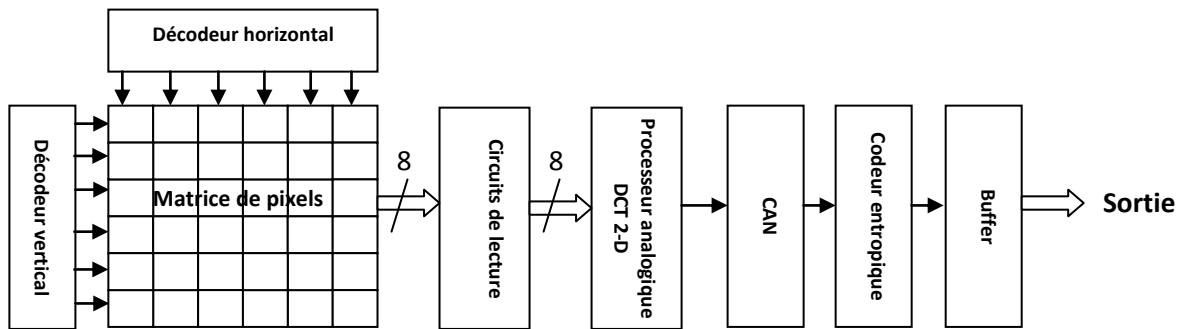


Figure 2.18 : Imageur CMOS intégrant un codeur à base de la DCT-2D [68]

Edwin J. Tan et al ont proposé dans [69] une méthode efficace et pratique pour implémenter la DCT-2D sur le plan focal de leur imageur CMOS. La technique utilisée consiste à employer la *Transformée de Fourier Arithmétique (AFT : Arithmetic Fourier Transform)* afin de calculer la DCT-2D. Cette approche est 5 fois moins complexe que la DCT classique, mais elle n'a été étudiée que théoriquement. La fabrication et le test du circuit sont nécessaires pour valider l'approche.

Abhishek Bandyopadhyay et al [70] présentent une architecture programmable intégrant un imageur CMOS et un processeur analogique flexible qui permet d'effectuer des transformations multiples sur une image acquise (DCT, ondelettes de Haar, etc.). L'utilisation de transistors à grille flottante facilite l'application des traitements bidimensionnels de type $Y = A^T \cdot P \cdot A$ (Y est l'image ou le bloc transformée, P est l'image ou le bloc de départ et A est la transformée à appliquer sur l'image ou sur le bloc). L'architecture proposée permet d'effectuer la première multiplication au niveau du pixel ($A^T P$) ; la deuxième multiplication est faite au niveau d'une matrice de multiplieurs à base de transistors à grille flottante. Cette architecture a été configurée deux fois : la première configuration pour appliquer sur l'image acquise la transformée en ondelettes de Haar, alors que la deuxième configuration pour appliquer la DCT-2D. L'image reconstruite est de bonne qualité ce qui prouve que l'architecture programmable proposée est bien fonctionnelle.

Ashkan Olyaei et Roman Genov présentent dans [71] leur capteur d'image CMOS fabriqué et validé en technologie CMOS standard 0.35 μm . Il s'agit d'un circuit qui comprend une matrice de pixels actifs, une unité pour la suppression du bruit spatial fixe (CDS), des convertisseurs Analogique/Numérique SIGMA-DELTA 8-bits au pied de la colonne de l'imageur (1 CAN/colonne), et un processeur dédié afin d'exécuter la transformée en ondelettes de Haar. Le processeur permet d'effectuer plusieurs niveaux de décompositions. Les résultats obtenus montrent que la qualité de l'image reconstruite est moyenne (un PSNR qui varie entre 16.29 dB et 24.92 dB avec un taux de compression variant de 8.55 à 12.04).

Canaan S. Hong et Richard Hornsey présentent dans [72] leur prototype de capteur d'image CMOS fabriqué en technologie standard $0.35\ \mu\text{m}$. Il s'agit d'une rétine intégrant des processeurs de traitement d'image au pied de la colonne de la matrice. Les traitements qui peuvent être appliqués sont multiples. En effet, le circuit peut fonctionner soit en mode normal (acquisition d'une image telle qu'elle est), soit en mode binaire (binarisation de l'image : comparer la valeur de chaque pixel à un seuil prédéfini et lui attribuer la valeur binaire '0' ou '1', c'est-à-dire attribuer le niveau de gris "blanc" ou "noir"). En plus, l'ensemble des processeurs au pied de la colonne de la matrice, permet d'appliquer des filtres d'ordre 3×3 sur l'image afin de calculer l'un des opérateurs suivants : "Erosion", "Dilatation" ou "détection de contour".

Evgeny Artyomov et al détaillent dans [73] le fonctionnement d'un imageur CMOS multi-résolution. Il s'agit d'une architecture flexible permettant l'acquisition d'images soit en mode pleine résolution (résolution originale) soit en mode faible résolution. Le passage en mode faible résolution est obtenu en calculant la moyenne d'un bloc de taille 2×2 ou 4×4 pixels. La rétine conçue est fabriquée en technologie standard $0.35\ \mu\text{m}$; elle comporte une matrice de pixels actifs, un décodeur ligne et un décodeur colonne pour la sélection d'un bloc de $N \times N$ pixels, deux échantillonneurs bloqueurs à 16 circuits chacun (pour échantillonner tout un bloc de 2×2 ou 4×4 pixels simultanément), un circuit de commutation permettant de commuter d'un échantillonneur bloqueur à l'autre, un multiplexeur permettant de multiplexer les sorties des deux échantillonneurs bloqueurs, un convertisseur Analogique/Numérique unique pour la numérisation des échantillons de l'un ou l'autre des deux échantillonneurs bloqueurs, et enfin une unité de traitement pour calculer la moyenne des valeurs numérisées (moyenne d'un bloc de $N \times N$ pixels). L'intérêt d'avoir deux échantillonneurs bloqueurs est le suivant : lorsque l'un effectue l'échantillonnage d'un pixel, l'autre envoie la valeur d'un autre pixel (appartenant à un autre bloc) déjà échantillonné vers le convertisseur A/N.

Sabrina E. Kemeny et al [74] ont aussi conçu et fabriqué un capteur d'image CMOS multi-résolution. Le passage du mode haute résolution au mode faible résolution se fait au pied de la colonne de la matrice en utilisant un réseau de capacités commutées. Plusieurs résolutions sont possibles : une fois que l'utilisateur choisit la taille des blocs à traiter (2×2 , 3×3 , 4×4 , etc.), le réseau de capacités commutées commence l'opération de moyennage pour chaque bloc de pixels. Le circuit, fabriqué en technologie standard $1.2\ \mu\text{m}$ de hp, consomme 5 mW pour une matrice de taille 128×128 pixels et une cadence d'images de 30 images/seconde.

Andrea Pinna nous présente dans [75] un nouveau système de vision sur puce. Il s'agit d'une rétine composée par un capteur d'image CMOS, une électronique de conversion Analogique/Numérique (un seul CAN pour toute la matrice), et une architecture parallèle spécialisée pour la simulation de réseaux connexionnistes (réseaux de neurones) : il s'agit d'un processeur neuronal capable de simuler en temps réel différents types de réseaux de neurones.

Abdelhafidh ELOUARDI et al proposent dans [76] une solution permettant de résoudre le problème de complexité de traitement du flux de données sortant d'un capteur d'image CMOS. La solution consiste à exécuter quelques traitements bas niveau sur le plan focal du capteur. Le système de vision conçu (PARIS1 : Programmable Analog Retina-like Image Sensor) intègre un vecteur de processeurs mixtes Analogiques/Numériques qui permet d'augmenter la vitesse d'exécution des calculs tout en réduisant les flux de données échangés avec un processeur numérique externe. Ce dernier aura donc comme entrées des informations numériques raréfiées qu'il pourra les exploiter sans aucun pré-traitement. Cette propriété a été prouvée en assemblant l'imageur PARIS1 avec un processeur ARM ; quelques traitements d'image standards (convolution, filtrage linéaire, détection de contours et détection de mouvement) ont été implémentés sur le nouveau système afin de valider la nouvelle architecture PARIS1-ARM.

Stuart Kleinfelder et al présentent dans [77, 78] leur imageur CMOS de résolution 352x288 pixels numériques. L'imageur, fabriqué en technologie CMOS standard 0.18 μm , est dédié à l'imagerie rapide. Le capteur a pu atteindre une vitesse d'acquisition de 10.000 images/seconde. Ceci n'est possible que par l'emploi d'une architecture spécifique adaptée aux traitements rapides. En effet, Stuart Kleinfelder a conçu son propre pixel numérique à 37 transistors. Chaque pixel inclut une photodiode, l'électronique nécessaire pour la transduction éclairissement-tension (4 transistors), un comparateur pour la conversion Analogique/Numérique 8 bits (9 transistors), et une mémoire dynamique pour le stockage de l'information numérisée (3 transistors / bit). L'imageur conçu possède 4 modes de fonctionnement : le premier est le mode standard (reset-intégration-CAN-lecture) ; le deuxième mode est le mode CDS permettant une lecture au début de la phase d'intégration et une autre lecture à la fin de cette phase pour remédier au problème du bruit spatial fixe (reset-CAN-lecture-intégration-CAN-lecture) ; le troisième mode est le mode « multiple sampling » ou « échantillons multiples » permettant d'atteindre une vitesse d'acquisition d'images élevée (reset-intégration-CAN-lecture-intégration-CAN-lecture...) ; et enfin le dernier mode qui permet d'avoir la vitesse

d'acquisition la plus élevée (lors de l'intégration d'une trame, le pixel effectue une lecture de la trame précédente). Les inconvénients majeurs de l'architecture de Stuart Kleinfelder sont : le faible taux de remplissage (15 %), la taille des pixels, et la consommation relativement élevée de l'imageur.

Jérôme Dubois et Dominique Ginhac ont présenté dans [55, 79] un travail de recherche qui a pour objectif la conception d'un système de vision dédié à l'imagerie rapide et aux traitements d'images. L'idée de départ était d'effectuer un traitement distribué sur toute la matrice de pixels. En effet, il s'agit d'une rétine qui inclut un ensemble de pixels numériques. Chaque pixel est formé par une photodiode, une électronique de transduction éclairement-tension, deux mémoires 8-bit, et un processeur élémentaire pour traiter l'image acquise. L'avantage de cette architecture distribuée est la rapidité d'acquisition et de traitement des images, mais l'inconvénient majeur est le nombre élevé de transistors par pixel (faible taux de remplissage, consommation énergétique et coût élevés). L'idée proposée par Jérôme Dubois et al. était donc d'utiliser un processeur élémentaire pour 4 pixels connexes afin de minimiser le nombre total de transistors et faciliter les traitements standards d'image (opérateur de Sobel, Laplacien, etc.). Une étude complémentaire a permis de déduire que l'utilisation d'une photodiode octogonale pour l'architecture retenue facilitera la connexion entre chaque processeur et les 4 éléments photosensibles qui l'entourent en topologie 4-connexe (figure 2.19). Pour la conversion Analogique/Numérique, Jérôme Dubois a opté pour un CAN/colonne. Afin d'avoir plus de flexibilité, l'architecture de l'imageur HISIC (High Speed Image Capture with processing at pixel level) est une architecture reconfigurable. En effet, HISIC peut être reconfiguré en fonction du type de traitement qu'on souhaite effectuer. En changeant les valeurs de 4 coefficients au niveau des processeurs, on passe d'un traitement à un autre (Sobel, Laplacien, gradient spatial, etc.) sans avoir recours à concevoir et fabriquer un imageur dédié pour chaque traitement.

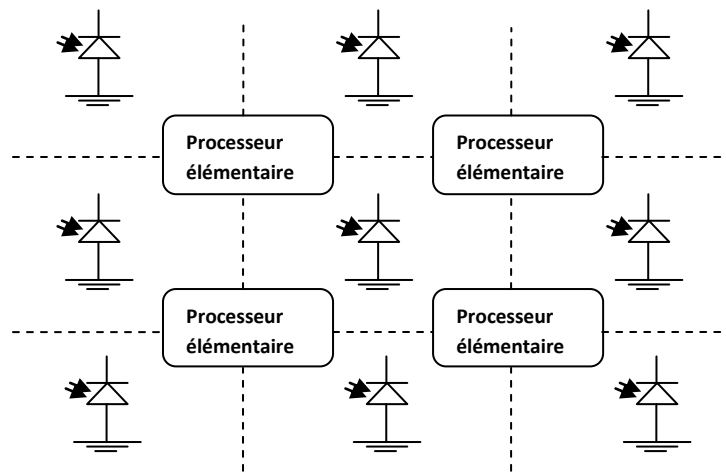


Figure 2.19 : Topologie 4-connexe [55]

Zhiqiang Lin et al ont proposé dans [80] une nouvelle technique de décomposition d'image. La technique donne des résultats comparables à la transformée en ondelettes discrète (Discrete Wavelet Transform), mais elle utilise une méthode différente. Celle-ci consiste à prédire la valeur d'un pixel en se référant aux valeurs de ses voisins. Une fois la valeur d'un pixel est prédite, un calcul de résidu (différence entre la valeur réelle du pixel et la valeur prédite) est fait. La valeur du résidu est généralement proche de zéro, et donc elle est codée sur un nombre de bits optimal ce qui induit à un taux de compression élevé. Chaque pixel est conçu de manière à pouvoir communiquer avec ses voisins. Il doit être capable à la fois de lire les valeurs des pixels voisins et d'envoyer sa propre valeur à son entourage. Les traitements sont effectués sur le plan focal de l'imageur, au niveau des pixels de la matrice. Les calculs qui devraient être faits (calculs de moyennes et de différences) nécessitent une architecture de pixels intégrant des capacités (mémoire de stockage) et des capacités commutées (calcul de moyenne ou de différence). Chaque pixel est conçu en fonction de sa position dans la matrice : les pixels de la sous-bande basse-fréquences du niveau de décomposition le plus élevé (sous bande la plus élémentaire, en haut à gauche) sont des pixels à 5 transistors et 1 capacité ; les pixels des autres sous-bandes sont des pixels à 14 transistors (à l'exception des sous-bandes diagonales dont les pixels incluent 15 transistors / pixel) et 4 capacités. L'imageur CMOS, de taille 33 x 25 pixels, a été conçu en technologie CMOS standard 0.5 μm . Le circuit consomme 0.25 mW pour une vitesse d'acquisition de 30 images/seconde. Les inconvénients majeurs de ce travail de recherche sont :

- La grande taille du pixel (69 μm x 69 μm).
- Le taux de remplissage relativement faible (20.7 %).
- Architecture non-uniforme des pixels de l'imageur, et par conséquent la non-uniformité du bruit au sein du capteur.

Qiang Luo et John G. Harris ont proposé dans [81] de concevoir un capteur d'image CMOS intégrant une version modifiée de la transformée en ondelettes de Haar (figure 2.20). La technique consiste à diviser l'image en blocs de taille 2x2 pixels et appliquer la transformée sur chaque bloc à part. Un réseau de capacités commutées au pied de la colonne de l'imageur est utilisé pour traiter simultanément tous les blocs d'un niveau, avant de passer au niveau suivant (les blocs situés en dessous). La technique CDS a été utilisée au pied de la colonne de l'imageur (avant de procéder à la transformation en ondelettes de Haar) afin de supprimer le bruit spatial fixe, le bruit en 1/f et le bruit thermique. Les coefficients résultants de cette transformation seront comparés à un seuil prédéfini ; uniquement les coefficients significatifs (> seuil) devraient être pris en compte pour un post-traitement externe. Un prototype de taille 4x4 pixels a été conçu et fabriqué en technologie standard 0.5µm.

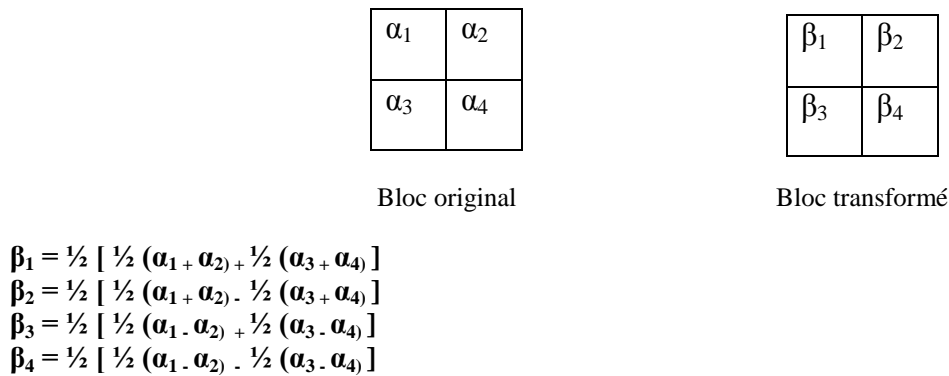


Figure 2.20 : Transformation en ondelettes de Haar modifiée [81]

K. Aizawa et al présentent dans [82] une nouvelle architecture de capteurs de vision CMOS permettant de détecter les pixels dont la valeur change au cours du temps. La valeur courante d'un pixel est comparée à l'ancienne valeur, déjà stockée en mémoire. Si la variation excède un seuil prédéfini, la nouvelle valeur devrait être envoyée à l'extérieur de l'imageur. Deux approches ont été étudiées.

- La première approche consiste à effectuer tout le traitement au sein du pixel. Le stockage et la comparaison des valeurs du pixel aux instants " t " et " $t - 1$ " se font au niveau du pixel. Cette approche nécessite l'emploi de 34 transistors par pixel.
- La deuxième approche consiste à faire tout le traitement au pied de la colonne de l'imageur. K. Aizawa et al ont utilisé un comparateur par colonne pour optimiser la surface du capteur.

Hawraa Amhaz a développé dans [83, 84, 85] plusieurs solutions visant à contrôler le flot de données en sortie de l'imageur afin de réduire les redondances spatiales et temporelles

des pixels. La première solution proposée consiste à comparer, au niveau de chaque ligne de la matrice, chaque pixel avec celui qui le précède; les pixels redondants ne subiront pas une conversion analogique/numérique, et seront représentés par un bit (flag) indiquant que le pixel courant porte la même valeur du pixel précédent. Cette solution permet uniquement de supprimer la redondance spatiale au niveau de chaque ligne de l'imageur; l'approche proposée néglige à la fois les redondances temporelles et les redondances spatiales qui considèrent tout le voisinage du pixel.

La deuxième solution proposée dans [85] permet d'exploiter les redondances temporelles. L'approche consiste à:

- Subdiviser la matrice de pixels en sous-blocs de taille $N \times N$.
- Calculer, pour chaque sous-bloc de l'imageur, la valeur moyenne des niveaux de gris de ses pixels.
- Stocker les valeurs moyennes des sous-blocs des images "i" et "i+1" dans des mémoires externes à la matrice de pixels.
- Comparer les valeurs moyennes des sous-blocs des deux images "i" et "i+1". Si la valeur moyenne d'un sous-bloc reste inchangée entre deux trames (images) successives, aucun traitement n'est envisagé. Dans le cas contraire, tous les pixels du sous-bloc seront scannés pixel par pixel afin de mettre à jour l'image courante.

L'inconvénient majeur de cette deuxième approche est le fait qu'elle exploite uniquement la redondance temporelle.

Alireza Nilchi et al proposent dans [86] un capteur d'image CMOS intégrant un processeur de traitement par colonne (figure 2.21). Chaque processeur intègre un bloc de signe, un multiplieur analogique, un accumulateur et un MADC (Multiplying Analog to Digital Converter). Ce dernier bloc joue le rôle d'un CAN et d'un multiplieur à la fois, permettant de multiplier la valeur du pixel par un coefficient numérique reconfigurable afin d'effectuer la transformation désirée sur l'image entière ou sur un bloc de l'image. Un prototype de taille 128x128 pixels a été fabriqué en technologie CMOS standard 0.35 μm . L'architecture a été validée par deux applications : la transformée en ondelettes de Haar et le calcul de différence entre deux trames successives.

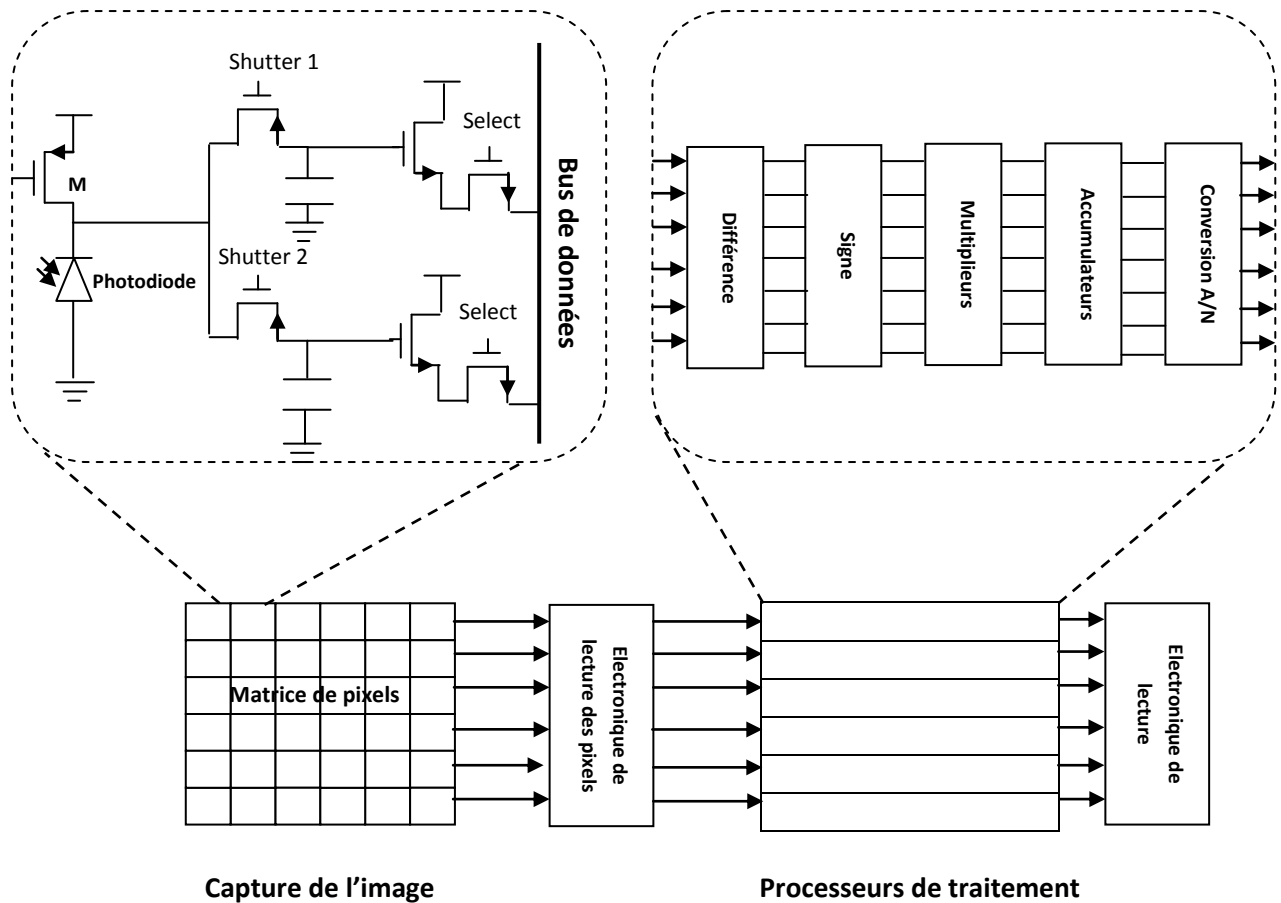


Figure 2.21 : Architecture de l'imageur présenté dans [86]

Yu M. Chi et al nous présentent dans [87] leur système de vision, capable de détecter un événement qui vient de se produire. La détection de l'événement se fait sur le plan focal de l'imageur grâce à une architecture spécifique de pixels. En effet, chaque pixel est capable de comparer sa valeur courante avec l'ancienne valeur, déjà stockée en mémoire. Un changement significatif de la valeur de l'intensité lumineuse de chaque pixel devrait être signalé à une unité de traitement externe (un microcontrôleur). Les blocs de l'image qui ont changé au cours du temps devraient être traités par le microcontrôleur. Le traitement consiste à appliquer la DCT-2D sur les blocs en question. Le système de vision conçu consomme 225 mW en cas de détection d'un événement et 6 mW en mode veille, mode durant lequel aucun événement n'est détecté.

Masatoshi Nishimura et Jan Van Der Spiegel présentent dans [88] un capteur d'image CMOS intégrant des techniques de reconnaissance de formes. Ces techniques sont basées sur la détection d'arrêt de ligne (line stops), la détection de coins et la détection de jonctions (de types X, Y et T). La combinaison de ces techniques, appliquées pour plusieurs plans d'orientation, permet de détecter les formes recherchées. L'algorithme de reconnaissance d'objets utilisé (basée sur les techniques citées précédemment) s'applique sur des blocs de

3x3 pixels. Pour cette raison, chaque pixel est connecté à tous ses voisins (3, 5 ou 8 voisins selon l'emplacement du pixel dans la matrice). Chaque pixel inclus un photo-détecteur et une unité de traitement reprogrammable, en fonction de la nature du calcul à faire. L'architecture massivement parallèle de l'imageur permet d'accélérer le traitement, mais la taille et la complexité du pixel sont trop élevés (147 transistors / pixel).

Malheureusement, tous les travaux de recherche cités ci-dessus présentent un même inconvénient qui rend inefficace leur éventuelle utilisation avec les réseaux de capteurs sans fil. En effet, ces travaux ont été développés dans des contextes différents de celui des réseaux de capteurs de vision sans fil. Les systèmes de vision conçus n'utilisent pas des techniques de traitement et de compression d'image adéquates pour les réseaux de capteurs sans fil (taux de compression élevé et complexité de calcul et coût raisonnables). L'émetteur/récepteur radio est le bloc le plus gourmand en énergie au sein d'un nœud de RCSF. Par conséquent, l'intégration, au sein des nœuds du réseau, d'un imageur intelligent implémentant des fonctionnalités de compression, à la fois efficaces et à faible complexité, pourrait faire face aux contraintes strictes de ce type de réseaux.

2.4. Conclusion

Dans ce chapitre, nous avons détaillé le fonctionnement et les caractéristiques des capteurs d'image CMOS. La nécessité de concevoir des systèmes de vision à faible consommation énergétique nous a poussé à détailler les différentes techniques permettant d'optimiser la consommation de l'imageur. Vu que la compression de données est la technique de conservation de l'énergie la plus efficace dans notre cas d'étude (conception d'un capteur d'image CMOS dédié aux RCSFs), nous avons réservé une partie pour présenter l'état de l'art concernant les capteurs d'image CMOS intégrant des fonctionnalités de compression sur le plan focal. Nous allons prouver dans les chapitres suivants l'intérêt de cette approche (conception d'un imageur intégrant des fonctionnalités de compression) dans un contexte de RCSFs.

Chapitre 3

Techniques de compression d'images et de vidéos adaptées aux réseaux de capteurs de vision sans fil

Introduction

Des études faites par Kenneth Barr et Krste Arsanovic [89] ont montré que le transcepteur (émetteur / récepteur) radio est le bloc qui consomme le plus d'énergie au sein d'un nœud de réseau de capteurs sans fil. La puissance dissipée lors de l'émission d'un seul bit peut servir pour exécuter quelques centaines voire quelques milliers d'instructions. La compression de données s'avère indispensable pour les nœuds capteurs fortement contraints en énergie et en mémoire, en particulier pour des applications multimédias caractérisées par un flux important de données à traiter et à transférer. Une attention particulière doit être prise en compte lors du choix de la technique de compression adaptée à ce type de réseaux (réseaux de capteurs de vision sans fil) : l'énergie consommée lors de la compression et la transmission de l'image compressée ne doit pas excéder l'énergie dissipée suite au traitement et au transfert de l'image non compressée. Le codeur implémenté doit répondre d'une part au compromis "complexité algorithmique – ressources accordées" et d'autre part au compromis "ressources accordées – qualité de l'image reconstituée".

Avant la phase de la conception matérielle de notre système de vision CMOS intégrant des fonctionnalités de compression, nous avons analysé les caractéristiques des codeurs envisagés pour implémentation. Nous nous sommes intéressés en particulier à l'étude du PSNR, taux de compression, distorsions de l'image, et la puissance dissipée. L'objectif de ce chapitre est donc de sélectionner les techniques de compression d'images et de vidéos qui répondent le mieux aux contraintes spécifiques des RCSFs [90, 91]. La plateforme de réseaux de capteurs sans fil *MICA2* [92] a été utilisée pour démontrer la validité de nos propositions en termes de complexité algorithmique, énergie consommée et qualité des images reconstruites.

3.1. Nécessité de la compression pour les réseaux de capteurs de vision sans fil

La motivation d'implémenter et d'étudier les performances de quelques techniques de compression d'image et de vidéo pour les réseaux de capteurs sans fil était basée sur une étude antérieure concernant l'implémentation, au sein du "*nœud source*", d'un algorithme qui permet l'acquisition, la paquetsation et la transmission d'une image entière sans compression à travers le réseau. La consommation énergétique du "*nœud source*" lors du traitement et du transfert de l'image non compressée (figure 3.1) est un critère décisif pour appliquer ou non la compression sur les images acquises.

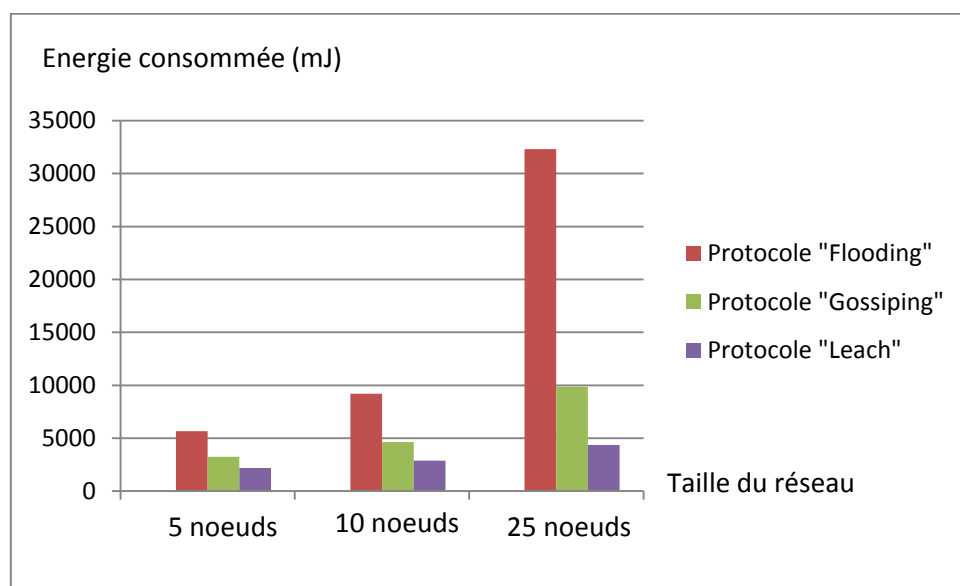


Figure 3.1 : Consommation énergétique du nœud source (plateforme MICA2) lors du traitement et de la transmission d'une image entière sans compression (image de taille 128x128 pixels codés sur 8 bpp)

Trois protocoles de routage différents ont été utilisés afin de transmettre l'image du "*nœud source*" jusqu'au "*nœud puits*" à travers le reste du réseau. Nous avons aussi fait varier la taille du réseau de capteurs sans fil pour voir l'impact du nombre de nœuds du réseau sur la consommation du "*nœud source*". Les protocoles de routage utilisés sont les suivants :

- **Protocole "Flooding" ou inondation**

Ce protocole consiste à diffuser chaque message dans tout le réseau. Chaque nœud qui reçoit un message pour la première fois diffuse l'information à tous ses voisins. Les voisins rediffusent aussi les paquets reçus pour la première fois à leur entourage direct. Ainsi, le message inonde le réseau jusqu'à atteindre la station de base.

- **Protocole "Gossiping"**

C'est une version légèrement améliorée de l'inondation. Le principe du protocole est que chaque nœud choisisse au hasard un seul voisin pour lui envoyer le message.

- **Protocole "LEACH" (Low Energy Adaptive Clustering Hierarchy)**

Proposé par Heinzelman et al dans [93], l'algorithme de clustering et de routage distribué nommé *LEACH* choisit de façon aléatoire les nœuds "*cluster-heads*" (un *cluster-head* est un nœud qui permet d'assurer la coordination entre les nœuds de son regroupement et le reste du réseau). Le rôle de "*cluster-head*" est attribué aux différents nœuds du réseau selon la politique de gestion "*round-robin*" (technique permettant d'attribuer à chaque nœud du réseau un intervalle de temps pendant lequel il joue le rôle de *cluster-head*) afin de garantir une dissipation équitable d'énergie entre les nœuds du réseau. Les nœuds appartenant à un "*cluster-head*" communiquent uniquement avec ce dernier alors que les "*cluster-heads*" communiquent ensemble afin de router les messages vers le *nœud puits*. Ce protocole est exécuté en deux phases. La première phase, appelée "*set-up*", consiste à sélectionner les nœuds dits "*cluster-heads*" et former les "*clusters*" de nœuds associés. La deuxième phase consiste à transmettre les données vers la station de base [15].

La figure 3.1 montre que la consommation énergétique augmente de façon considérable avec la taille du réseau. On a donc intérêt à mettre en place des réseaux de petite taille. La figure montre aussi clairement la nécessité d'exclure le protocole "*Flooding*" qui a une consommation énergétique trop élevée par rapport aux deux autres techniques de routage. La valeur importante de l'énergie dissipée est due au fait de recevoir un grand nombre de paquets de la part des nœuds voisins au nœud source. Ce dernier reçoit à plusieurs reprises les mêmes messages en provenance de plusieurs nœuds du réseau.

La consommation du nœud source est relativement améliorée dans le cas du protocole "*Gossiping*" par rapport au protocole "*Flooding*" puisque le nombre de paquets reçus par ce nœud devrait baisser de façon considérable. En effet, les nœuds voisins du nœud source choisissent un seul nœud de leurs voisinages afin de transmettre leurs messages ; le nœud source n'est pas forcément sélectionné à tous les coups comme destination. Malgré ce gain, l'énergie dissipée reste trop élevée. Le nœud dissipe 9213 mJ lors du traitement et du transfert d'une image de taille 128x128 pixels (8 bits par pixel) pour un réseau qui inclut 25 nœuds.

Le protocole LEACH consomme peu d'énergie par rapport aux deux autres protocoles étudiés. Ce protocole fait partie des techniques de routage (dédiées aux RCSFs) les plus appréciées par la communauté scientifique. La figure 3.1 montre que l'énergie dissipée varie peu en fonction de la taille du réseau. En plus, le gain atteint est très important par rapport aux deux autres techniques de routage présentées. L'énergie consommée par le nœud source est de 2190 mJ pour un réseau de 5 nœuds, 3250 mJ pour un réseau de 10 nœuds et 5668 mJ pour un réseau de 25 nœuds.

Malgré l'économie d'énergie apportée par le protocole LEACH, la consommation du nœud source reste relativement élevée pour un nœud capteur à contraintes énergétiques strictes. L'adoption d'une technique de compression efficace pourrait étendre d'une part la durée de vie du nœud source et d'autre part la durée de vie de la totalité du réseau [7]. Nous avons déjà présenté dans le chapitre 1 quelques travaux de recherche qui mettent en relief l'importance de la compression d'image pour les RCSFs. Dans la suite de ce chapitre, nous allons étudier quelques chaînes de compression d'image et de vidéo dans le but de justifier l'intérêt de la compression de données pour les RCSFs.

3.2. Approches de compression d'image pour les réseaux de capteurs de vision sans fil

Les techniques de compression d'image usuelles comme JPEG ou JPEG 2000 ont un coût énergétique bien supérieur au gain qu'ils apportent sur l'émetteur/récepteur radio. Le nœud capteur épuiserait plus vite son énergie en transmettant des images compressées que des images non compressées [94]. Le développement de nouvelles méthodes de compression d'image efficaces en énergie paraît être une nécessité pour rendre pratique le déploiement de réseaux de capteurs de vision sans fil [7].

Dans la littérature, différentes techniques de codage existent. Les techniques de compression basées sur les méthodes différentielles et prédictives sont très efficaces en terme de taux de compression, mais le coût énergétique de ces approches ne permet pas une éventuelle implémentation sur des nœuds de réseaux de capteurs à contraintes multiples. En effet, les techniques différentielles consistent soit à calculer et coder la différence entre deux pixels voisins soit à calculer et coder la différence entre les valeurs d'un même pixel entre deux trames successives. Ce traitement nécessite des mémoires tampons pour stocker ces différences et parfois stocker les valeurs des pixels de la trame précédente (si la technique

utilisée nécessite un traitement temporel). Le traitement est parfois beaucoup plus compliqué ; certaines techniques de codage différentiel utilisent à la fois un voisinage spatial du pixel (de taille $N \times N$ pixels) et un traitement spatio-temporel pour coder chaque pixel de l'image. En ce qui concerne les techniques prédictives, elles ont aussi une complexité élevée. Les approches prédictives consistent à estimer la valeur d'un pixel par rapport à son voisinage spatial ou spatio-temporel ; l'erreur d'estimation (différence entre la valeur estimée et la valeur réelle du coefficient à coder) serait par la suite codée. Pour conclure, les standards classiques de compression d'image et de vidéos (JPEG, JPEG2000, etc.) ainsi que les techniques de codage différentielles et prédictives sont loin d'être employées dans un contexte de réseaux de capteurs sans fil. Par conséquent, nous avons choisi d'étudier quelques techniques de compression hybrides basées sur le codage par transformation (transformées en cosinus discrètes et en ondelettes). Ces approches (codage par transformation) sont très utilisées par les scientifiques puisqu'ils fournissent le meilleur compromis "*complexité - taux de compression - distorsion de l'image*".

Nous allons présenter dans la section suivante quelques techniques hybrides de compression d'image à complexité réduite. Ces techniques sont basées sur les transformées en cosinus discrète et les transformées en ondelettes [36]. Une comparaison entre les différents codeurs étudiés permettra de mettre en évidence l'approche à adopter pour une future implémentation matérielle.

3.2.1. Techniques basées sur la transformée en cosinus discrète (DCT)

Un codeur basé sur la transformée en cosinus discrète (figure 3.2) commence par le découpage de l'image en blocs de taille $N \times N$ pixels (8×8 ou 16×16 pixels). La chaîne de compression sera par la suite appliquée sur chaque bloc de l'image à part. Les blocs de l'image subissent tout d'abord une étape de transformation (DCT) suivie par une étape de quantification et une étape de codage (codeur entropique). Notre approche consiste à optimiser chaque bloc de la chaîne de compression étudiée sans trop dégrader les performances du codeur.

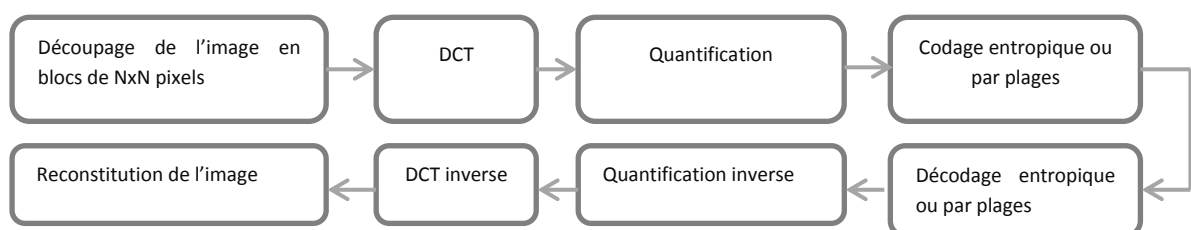


Figure 3.2 : Organigramme d'un Codeur/Décodeur basé sur la transformée en cosinus discrète

- **Transformée de l'image**

La DCT classique est trop gourmande en énergie. Elle consomme environ 60 % de l'énergie totale du codeur JPEG. La DCT de Loeffler (figure 3.3) et la DCT zonale (on parle de DCTs rapides) sont deux algorithmes moins complexes que la DCT classique, et qui peuvent être implémentés facilement sur des nœuds de réseaux de capteurs de vision sans fil [7, 38]. La DCT Cordic consomme uniquement 38 % de l'énergie requise par la DCT de Loeffler puisqu'elle remplace les multiplications (très coûteuses en énergie) par des additions et des décalages, mais l'image reconstruite subit une dégradation significative par rapport à la DCT de Loeffler. L'économie de l'énergie se fait donc au prix d'une distorsion importante de l'image [95].

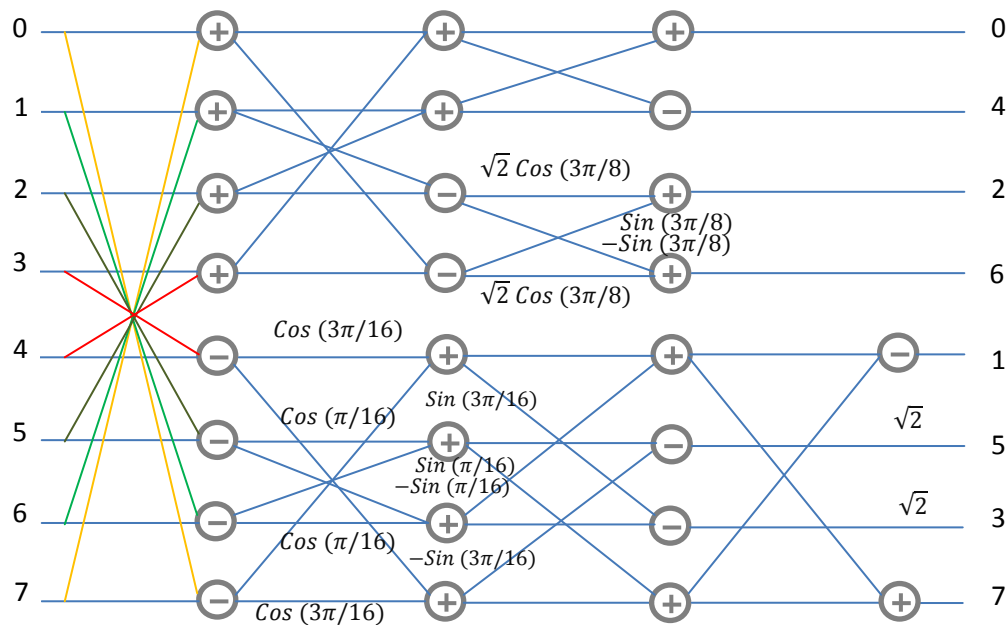


Figure 3.3 : Graphe des flux de la DCT de Loeffler

De toutes les approches de transformation citées, la DCT de Loeffler fournit le meilleur compromis entre la complexité de calcul et la distorsion de l'image [95, 96]. Pour cette raison, nous avons opté pour cet algorithme dans la chaîne de compression à tester.

- **La quantification**

La quantification est l'étape au cours de laquelle se produit la majeure partie de perte d'information, ce qui permettra par la suite de compresser efficacement les données au prix d'une dégradation de la qualité de l'image reconstruite. La quantification consiste à diviser chaque bloc (généralement de taille 8x8 coefficients) transformé (par application de l'une des

variantes de la transformée en cosinus discrète) de l'image par une matrice appelée *matrice de quantification* de même dimension (figure 3.4). Les coefficients de cette matrice permettent d'atténuer les hautes fréquences dont l'œil humain est très peu sensible. Les hautes fréquences sont représentées par les coefficients (résultants de l'application de la DCT sur les blocs de l'image originale) de faibles valeurs au sein de chaque bloc transformé. Les valeurs de ces coefficients seraient plus atténuées lors de l'application de l'étape de quantification. La plupart de ces coefficients sont ramenés à zéro par la matrice de quantification ce qui contribue fortement à une compression efficace de l'image.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figure 3.4 : Matrice de quantification JPEG pour la luminance [37]

- **Codeur entropique ou par plages**

Le codage entropique correspond à un codage statistique sans pertes des coefficients quantifiés avec pour objectif d'atteindre l'entropie. L'entropie (équation 3.1) caractérise la quantité d'information optimale que contient une source de données. En pratique, l'entropie d'une image est inversement liée à la probabilité d'apparition des coefficients dans une image ou une image transformée : une faible probabilité d'apparition d'un niveau de gris ou d'un coefficient contribue à une grande valeur de l'entropie globale et inversement.

$$\text{Equation 3.1} \quad H = - \sum_{n=0}^{M-1} Pr(X_n) \cdot \log(Pr(X_n)) \quad (bpp)$$

H: entropie

X_n: coefficient ou niveau de gris ayant la valeur X_n

Pr(X_n) : probabilité d'apparition du coefficient X_n

Plusieurs techniques de codage entropique existent dans la littérature. Certains codeurs sont appréciés grâce à leur efficacité (taux de compression élevé) alors que d'autres le sont pour leur complexité réduite. Le principe d'un codeur entropique est d'attribuer des codes plus courts aux symboles fréquents et de réserver des codes plus longs aux symboles peu fréquents. Le code d'un symbole ne doit pas être le préfixe d'un autre. Le calcul des

probabilités d'apparition des symboles à coder se fait soit de façon statique (avant de commencer le codage) soit de façon dynamique (au fur et à mesure du processus de codage). Dans le deuxième cas, on parle d'algorithmes adaptatifs caractérisés par une complexité algorithmique importante ; par conséquent, ce type d'algorithmes ne fait pas l'objet de notre étude.

L'algorithme de Shannon-Fanno, le codage de Huffman et le codage arithmétique sont les codeurs entropiques les plus connues. La technique de Shannon-Fanno est la moins complexe, mais elle est rarement optimale (technique optimale uniquement dans certains cas particuliers). Le codage de Huffman et le codage arithmétique sont plus performants, mais ce dernier est plus pratique. En effet, la technique de codage arithmétique, contrairement aux autres codeurs entropiques, permet de générer un code associé à la séquence entière et non pas un code associé à chaque symbole de la séquence à coder. Cette caractéristique a permis au codage arithmétique de surmonter les limites de performance de tous les codeurs statistiques existants au prix d'une complexité de calcul légèrement supérieure.

D'autres techniques de codage appelées "techniques de codage par plages" peuvent être exploitées. Les caractéristiques attirantes de ces codeurs sont la simplicité d'implémentation et la complexité assez réduite par rapport aux codeurs statistiques. L'idée de base ici est de chercher les suites des éléments répétitifs et de les coder sous la forme "coefficient à coder – nombre d'occurrences". Vu que les coefficients de l'image transformée et quantifiée contiennent plusieurs suites de "zéros" successifs, un codeur de type *RLE* (Run Length Encoding) peut contribuer à l'apport de résultats satisfaisants en termes de performances et de complexité de la chaîne de compression globale [36].

Pour résumer, les deux chaînes de compression d'image adoptées dans cette première partie (concernant les techniques à base de DCT) sont présentées dans la figure suivante :

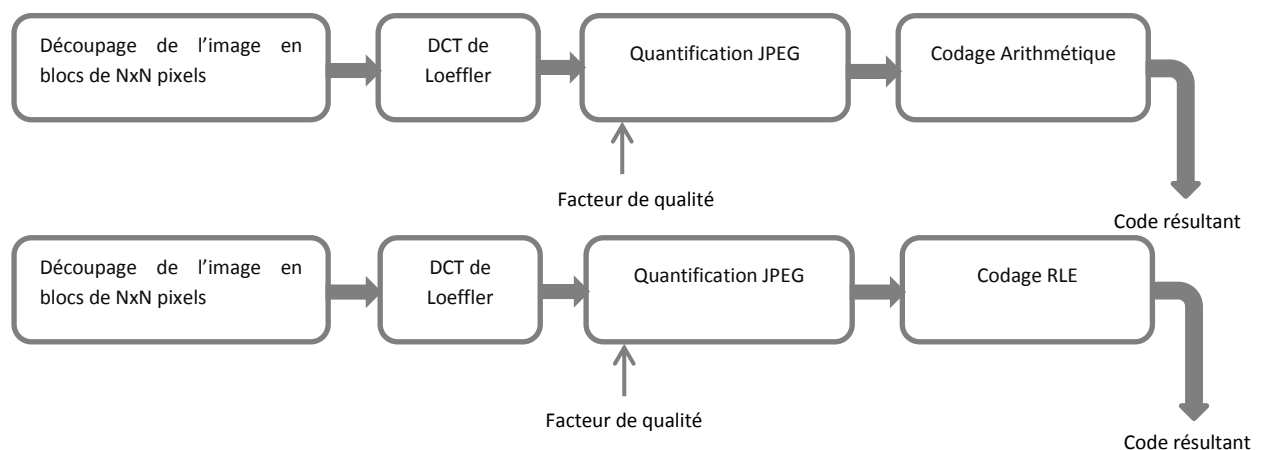


Figure 3.5 : Chaines de compression d'image à base de la transformée en cosinus discrète

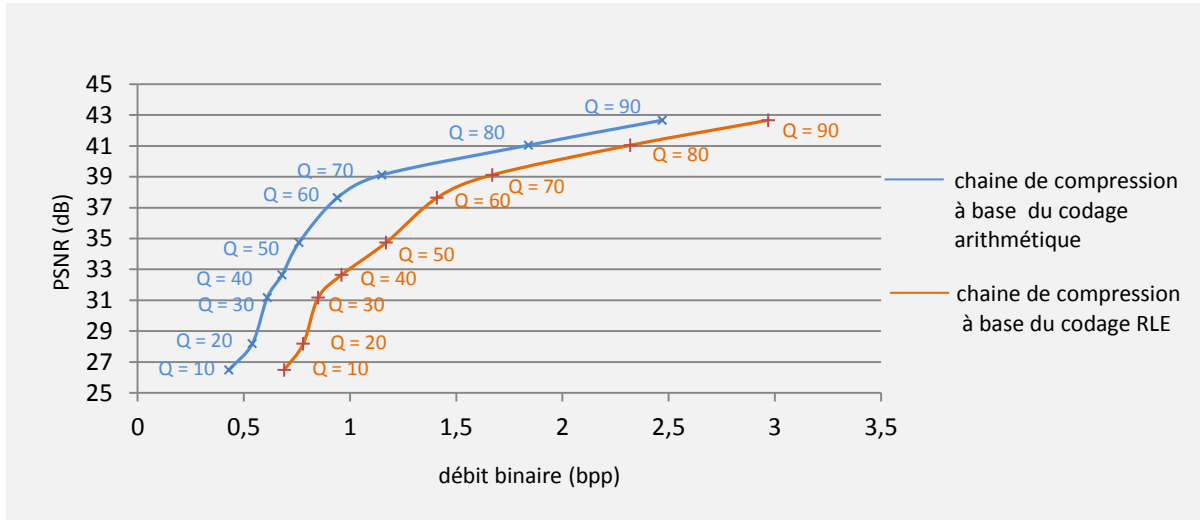


Figure 3.6 : Caractéristique "débit-distorsion" des codeurs étudiés

La figure 3.6 présente les performances des deux chaînes de compression d'image étudiées dans cette partie. Les métriques de performance utilisées sont :

- **Mesures de distorsion**

La distorsion est l'erreur introduite par l'opération de compression, due au fait que éventuellement l'image reconstruite n'est pas exactement identique à l'image originale. La mesure de distorsion utilisée généralement en compression d'image est l'erreur quadratique moyenne MSE (Mean Square Error) :

$$\text{Equation 3.2} \quad MSE = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I(m, n) - \hat{I}(m, n)]^2$$

M, N : définissent la taille de l'image à étudier

I(m, n) : pixel de coordonnées (m, n) de l'image originale

$\hat{I}(m, n)$: pixel de coordonnées (m, n) de l'image reconstruite

On référence cette erreur par rapport à la dynamique de l'image en décibels. On obtient un rapport signal sur bruit crête à crête noté *PSNR (Peak Signal to Noise Ratio)* :

$$\text{Equation 3.3} \quad PSNR = 10 \cdot \log_{10} [(2^R - 1)^2 / MSE] \text{ dB}$$

R : dynamique du signal

- **Taux de compression**

Le taux de compression T_C est défini par la relation suivante :

$$\text{Equation 3.4} \quad T_C = \frac{\text{nombre de bits de l'image originale}}{\text{nombre de bits de l'image comprimée}}$$

Le débit binaire *DB* (en bits par pixel ou bpp) est défini par la relation suivante:

$$\text{Equation 3.5} \quad DB = \frac{\text{nombre de bits qui codent chaque pixel de l'image originale}}{T_C}$$

La figure 3.6 montre que la chaîne de compression qui intègre un codeur arithmétique est plus performante que celle basée sur le codage *RLE*. En effet, le codage arithmétique offre un meilleur taux de compression à qualité d'image égale. La supériorité du codeur arithmétique est traduite par une complexité de calcul beaucoup plus élevée que celle du codage *RLE*. Par conséquent, l'écart de consommation entre les deux nœuds source qui exécutent ces deux techniques de compression est très important.

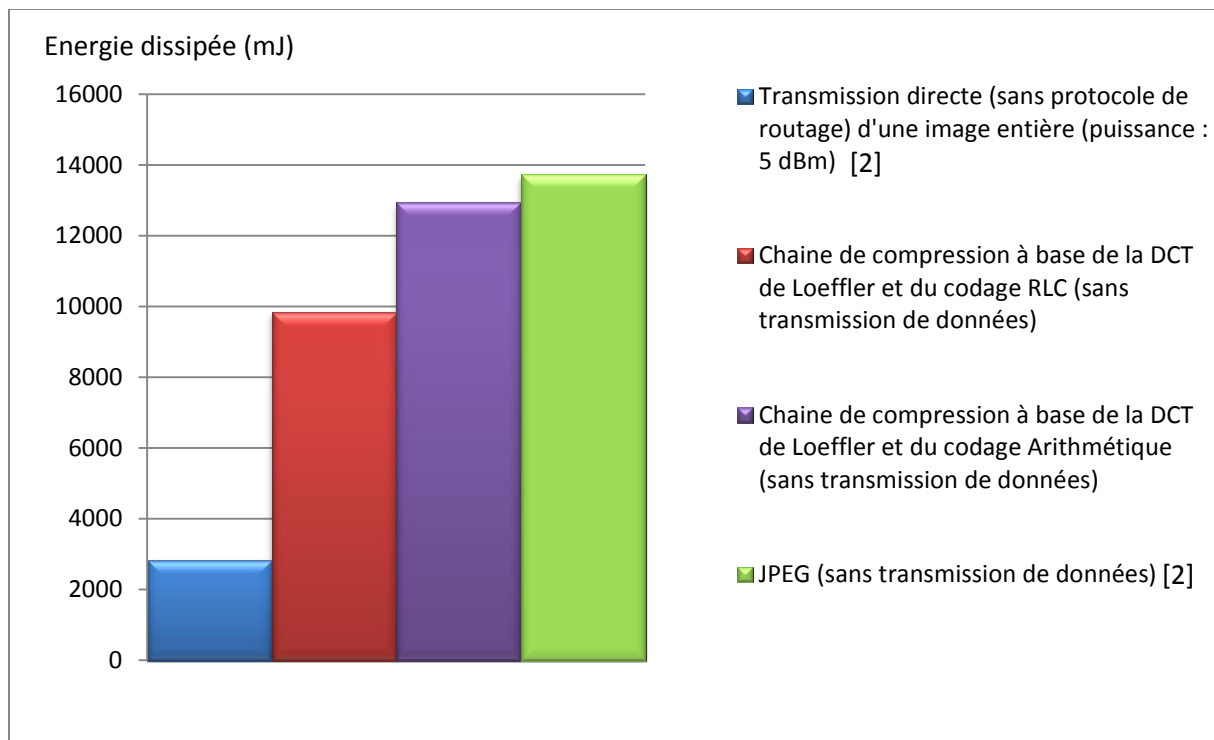


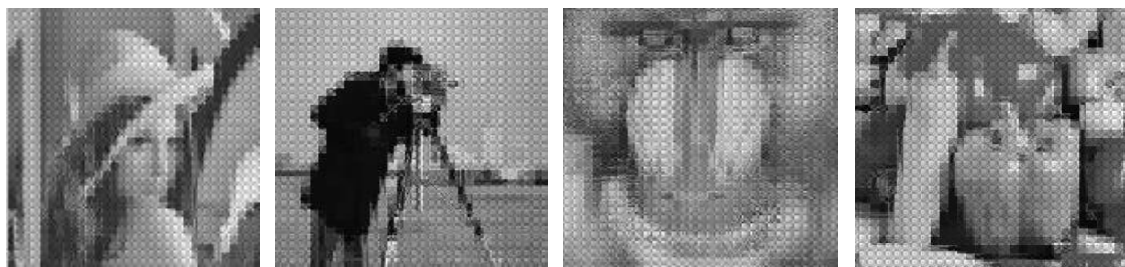
Figure 3.7 : Consommations typiques d'un nœud source (plateforme MICA2) pour des applications liées au traitement et à la transmission d'images (de taille 128 x 128 pixels codés sur 8 bpp) pour un facteur de qualité $Q = 97$

Les résultats présentés dans la figure 3.7 prouvent que les techniques de compression à base de la transformée en cosinus discrète (DCT) sont très coûteuses en énergie. L'énergie dissipée lors du traitement dépasse toujours celle consommée lors de la transmission directe d'une image non compressée. Malgré que la DCT de Loeffler ait une complexité de calcul relativement inférieure à celle de la DCT classique, les algorithmes de codage utilisés restent loin de satisfaire aux besoins spécifiques des nœuds capteurs strictement contraints en énergie. La figure nous montre aussi un écart considérable entre les consommations des deux codeurs étudiés (écart de 3093 mJ) ; cet écart s'explique par la complexité du codage arithmétique par rapport au codage RLE.

Une autre raison, pour laquelle les chaînes de compression basées sur la DCT ne sont pas appréciées, est les distorsions perçues des images reconstruites pour des taux de compression relativement élevés (figure 3.8).



Images originales



Images reconstruites (Facteur de qualité $Q = 20$)

Figure 3.8 : Distorsion des images reconstruites lors de l'application des chaînes de compression étudiées

Les consommations élevées des codeurs étudiés et les distorsions gênantes perçues sur les images reconstruites nous ont poussées à aller plus loin dans notre étude. Les techniques de compression basées sur les transformées en ondelettes peuvent être exploitées puisqu'elles permettent de reconstruire les images sans distorsions. En plus, ce type de transformations est peu coûteux en termes de complexité de calcul et de puissance consommée par rapport aux transformations en cosinus discrète. Cristian Duran-Faundez et al ont mesuré l'énergie consommée par un nœud *MICA2* lorsqu'il exécute la transformée en ondelettes discrète (*DWT : Discrete Wavelet Transform*) sur une image monochrome de taille 128x128 pixels codée sur 8 bpp. Le coût énergétique est de 151 mJ pour un seul niveau de décomposition et de 188 mJ pour deux niveaux de décomposition (l'algorithme est exécuté récursivement deux fois) [7]. La même équipe de recherche a montré que le même nœud capteur consomme 11264 mJ pour calculer la DCT classique pour une image ayant la même taille et les mêmes caractéristiques. Par conséquent, les techniques de compression à base d'ondelettes pourraient être adoptées s'ils fournissent un bon rapport "*débit-distorsion*" ainsi qu'un gain énergétique considérable par rapport aux chaînes de compression à base de la DCT et bien évidemment par rapport au transfert d'images non compressées.

Dans la partie suivante, nous allons étudier quelques chaînes de compression basées sur les transformées en ondelettes afin de trouver une solution adéquate aux réseaux de capteurs sans fil.

3.2.2. Techniques basées sur la transformée en ondelettes discrète

La transformée en ondelettes offre une meilleure finesse que la transformée en cosinus discrète (DCT). Elle permet, contrairement à la DCT, de ressortir les grandes variations et les détails les plus fins du signal analysé. Ceci est dû à une caractérisation en temps et en fréquence du signal traité.

La transformée en ondelettes discrète est une transformation qui décompose un signal numérique en deux parties en utilisant un filtre passe-bas " L " et un filtre passe-haut " H ". La composante résultant du filtrage passe bas est une représentation approximative du signal original ramené à la nouvelle résolution. L'autre partie ou composante représente les détails perdus en passant de la résolution originale (signal original) à la nouvelle résolution.

Vu qu'on traite des images numériques 2-D, la transformée en ondelettes est validée en appliquant les filtres " L " et " H " sur les pixels de l'image ligne par ligne. Le même traitement est appliqué sur les coefficients résultants, colonne par colonne cette fois-ci. Ainsi, on obtient une image transformée, formée par 4 sous-images ou sous-bandes (figure 3.9).

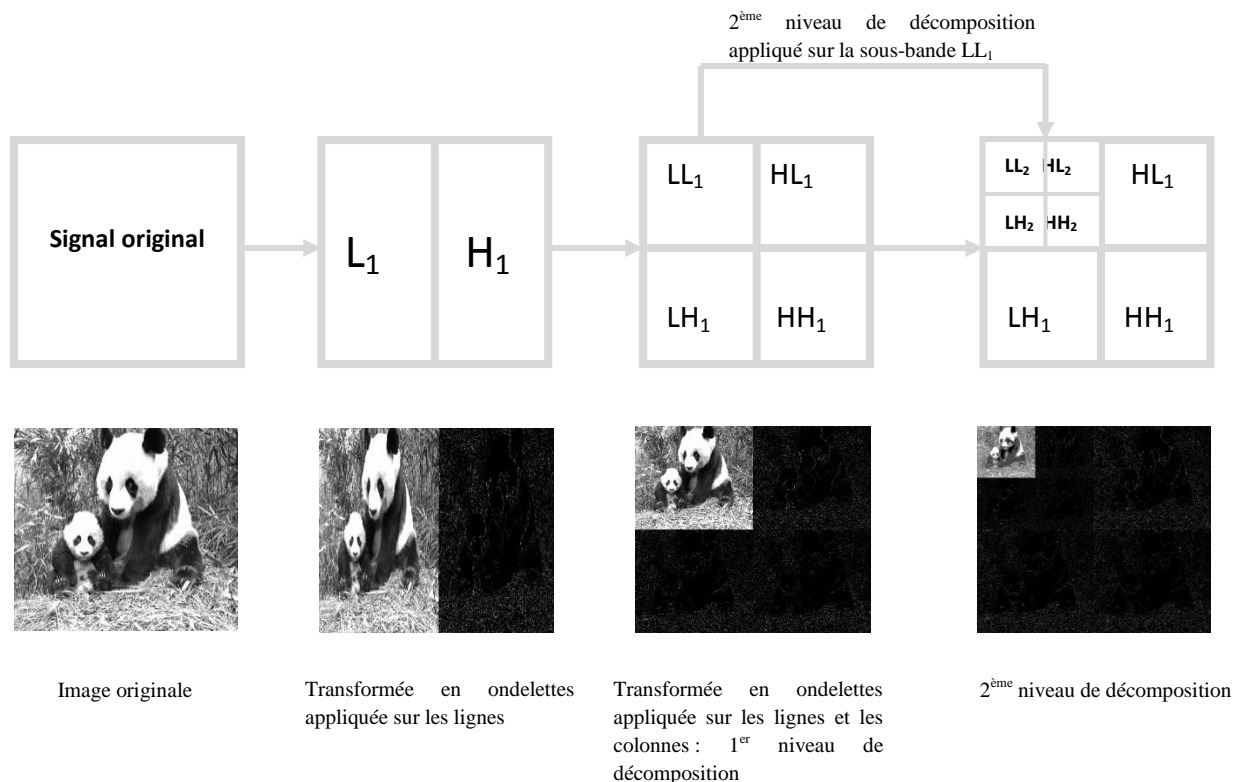


Figure 3.9 : Décomposition en ondelettes pour deux niveaux de décomposition

La sous-bande LL_1 représente une version à l'échelle $\frac{1}{2}$ de l'image originale. Les sous-bandes LH_1 , HL_1 et HH_1 fournissent les détails perdus respectivement dans les trois directions horizontale, verticale et diagonale. La transformée en ondelettes peut être appliquée récursivement sur la sous-bande LL_i du niveau de décomposition précédent. Ceci permettra d'obtenir plusieurs niveaux de décomposition au prix d'une perte de détails considérable [7, 97].

Plusieurs variantes de la transformée en ondelettes existent dans la littérature. Le choix de l'ondelette adaptée n'est pas aisé. Le bon choix consiste à bien cerner le problème (compression d'image, compression vidéo, détection de contours, etc.) ainsi que les besoins de l'application (précision de calcul, qualité du signal reconstruit, complexité du traitement, consommation, etc.) afin d'identifier le type de la transformée à utiliser. Donc, on ne peut pas parler d'ondelette parfaite ou idéale, adaptée à toutes les applications.

La figure 3.10 présente l'organigramme d'un Codeur/Décodeur basé sur la transformée en ondelettes:

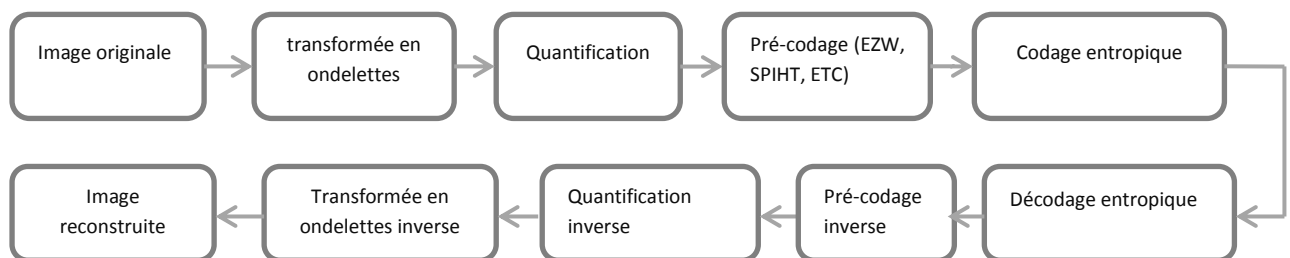


Figure 3.10 : Organigramme d'un Codeur/Décodeur basé sur la transformée en ondelettes

- ***La transformée en ondelettes***

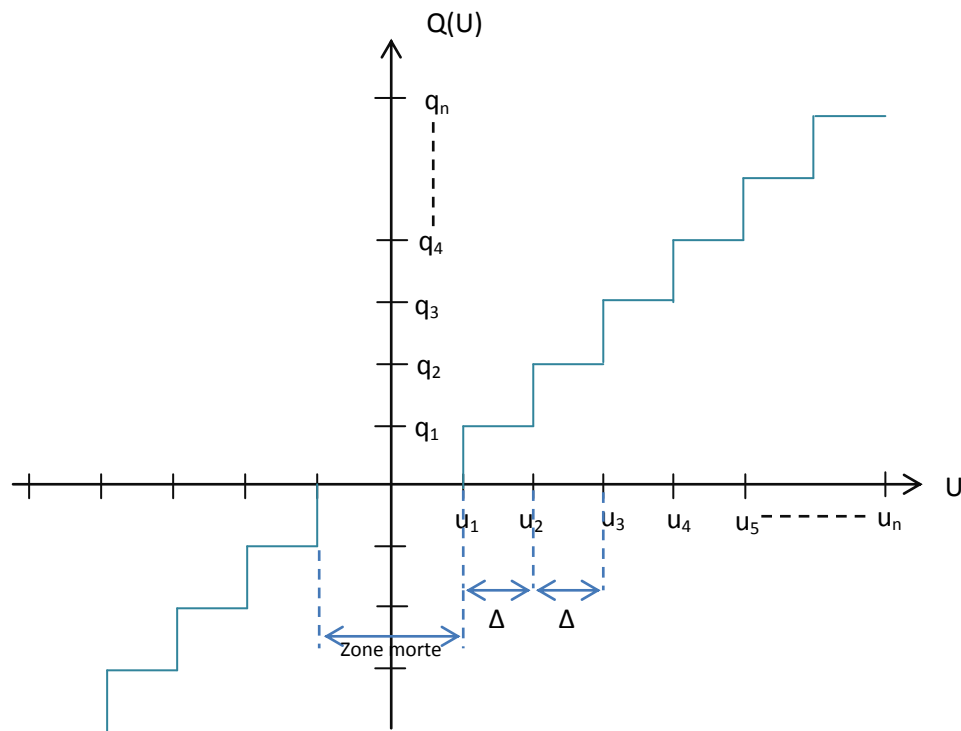
Rappelons que notre objectif est de chercher une approche de compression d'image à faible complexité. La technique adoptée devrait être implémentée sur une même puce avec un capteur d'image CMOS à faible consommation d'énergie. L'ensemble est dédié à une future implémentation matérielle sur un nœud de réseau de capteurs sans fil. La transformée en ondelettes de Haar est la moins complexe et la plus simple à implémenter, en comparaison avec toutes les autres ondelettes (ondelettes de Littlewood-Paley, de Daubechies, de Meyer, Battle-Lemarié, etc). En plus, le calcul de la transformée en ondelettes de Haar (transformée directe et transformée inverse) est très précis ce qui permettrait une reconstitution parfaite du signal original [97].

La transformée discrète en ondelettes de Haar, appliquée à un vecteur (ligne ou colonne) $f(x) = [y_1, y_2, y_3, \dots, y_n]$ génère un vecteur $[[a_1, a_2, a_3, \dots, a_{n/2}], [b_1, b_2, b_3, \dots, b_{n/2}]]$ avec :

- $[a_1, a_2, a_3, \dots, a_{n/2}]$: approximation des moyennes des valeurs de la fonction $f(x)$ prises deux à deux ($a_1 = (y_1 + y_2) / 2$, $a_2 = (y_3 + y_4) / 2$, etc.).
- $[b_1, b_2, b_3, \dots, b_{n/2}]$: approximation des coefficients de détails calculés de la façon suivante : $b_1 = y_1 - a_1$, $b_2 = y_2 - a_2$, etc.).

- ***La technique de quantification***

La performance d'un codeur basée sur la transformée en ondelettes dépend de l'étage de quantification utilisé. Le quantifieur (ou quantificateur) a pour rôle de réduire la proportion des coefficients significatifs et par conséquent le nombre de bits du code résultant en sortie du codeur. Le quantificateur employé a une influence majeure sur le taux de compression atteint par le codeur. La quantification peut être scalaire (appliquée sur chaque coefficient à part) ou vectorielle (appliquée sur un vecteur de coefficients). La quantification vectorielle nécessite un calcul intensif, ce qui n'est pas apprécié dans notre cas d'étude [36]. Un quantifieur scalaire a une complexité raisonnable et aboutit généralement à des performances satisfaisantes du codeur. Le quantifieur scalaire uniforme (le pas de quantification est fixe) à zone morte (chaque coefficient appartenant à cette zone ou intervalle est mis à zéro) est la technique de quantification la plus pratique pour les chaînes de compression à base d'ondelettes. Cette technique (figure 3.11) a prouvé son efficacité vu qu'elle a été adoptée dans la norme JPEG 2000, standard de compression d'image le plus puissant de nos jours.



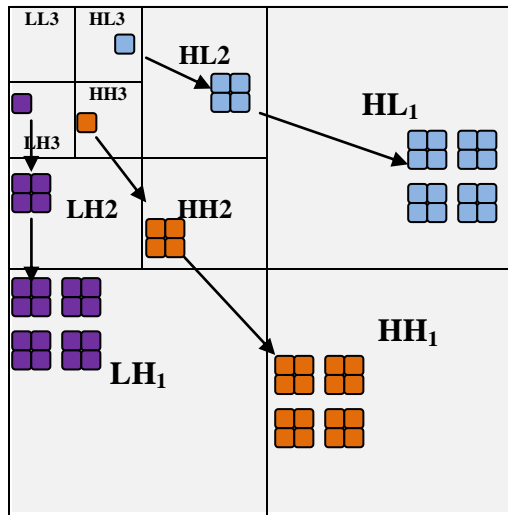
Pas de quantification : $\Delta = u_i - u_{i-1}$
 Valeur quantifiée : $Q(u_i) = \hat{u}_i = q_i$

Figure 3.11 : Quantifieur scalaire uniforme à zone morte

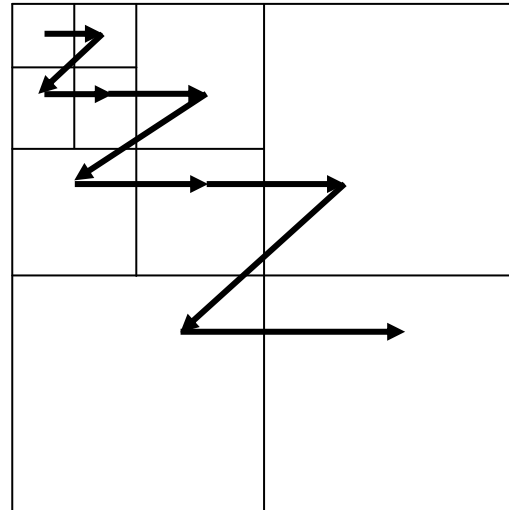
- **Choix de la technique de pré-codage**

Avant de procéder au codage (entropique ou à longueur variable), une étape de pré-codage est nécessaire. Cette étape permet de parcourir et coder les coefficients d'ondelettes en fonction de leurs degrés d'importance (d'où le nom "*codeurs progressifs*"). Les codeurs progressifs les plus connues sont *SPIHT* et *EZW*. *SPIHT* est une version améliorée de l'algorithme *EZW*. Il est plus performant qu'*EZW*, mais au prix d'une complexité de calcul relativement plus élevée. *EZW* répond mieux aux exigences des réseaux de capteurs sans fil, en termes de complexité de calcul et simplicité d'implémentation [36].

En général, dans une représentation d'image par coefficients d'ondelettes, l'image obtenue est organisée de façon à représenter les principaux traits de l'image dans les bandes de basses fréquences puis les détails dans les bandes de hautes fréquences. Le principe de l'algorithme *EZW* s'appuie sur cette représentation, pour coder les coefficients d'une manière progressive. Ainsi, on commence par les basses fréquences, ensuite on code les détails (hautes fréquences). L'avantage de cet algorithme, est que l'on a à tout moment un niveau de compression et que l'on peut arrêter à chaque instant le codage.



(a) représentation de l'organisation en arbre des coefficients d'ondelettes



(b) ordre prédéfini de parcours de l'image par l'algorithme EZW

Figure 3.12 : Exploitation des coefficients d'ondelettes par l'algorithme EZW

L'algorithme *EZW* comprend deux parties : une tâche principale (appelée aussi passage dominant) et une tâche secondaire ayant pour rôle le raffinement de la reconstruction de l'image pendant la phase de décodage.

Traitement principal

Le traitement principal commence par une phase d'initialisation permettant de calculer le seuil de départ " t_0 ", défini par l'équation suivante :

$$\text{Equation 3.6} \quad t_0 = 2^{\lfloor \ln(\max |c(x, y)|) \rfloor}$$

t_0 : le seuil initial

$c(x, y)$: désigne les coefficients d'ondelettes

Par la suite, la matrice est parcourue en "zigzag" comme le montre la figure 3.12 (b). Cette méthode de parcours a été choisie de manière à préserver l'ordre d'importance des coefficients traités. En effet, la technique parcourt les coefficients de basse fréquence en premier, puis permet d'avancer graduellement vers les détails (hautes fréquences).

Chacun des coefficients parcourus est comparé (en valeur absolue) au seuil " t_0 ". Si le coefficient est supérieur au seuil, il est codé "Positif" ou "Négatif" selon le signe ; sinon il est soit "Zero isolé" ou "Zerotree". On se ramène ainsi à coder K symboles avec un dictionnaire de quatre symboles (2 bits pour coder 4 symboles) :

- ✓ Positif (P) : indique que la valeur absolue du coefficient traité est supérieure au seuil et que son signe est positif.

- ✓ Négatif (N) : indique que la valeur absolue du coefficient traité est supérieure au seuil et que son signe est négatif.
- ✓ Zéro isolé (Z) : indique que la valeur absolue du coefficient traité est inférieure au seuil " t_0 " et qu'il existe parmi ses descendants (selon l'arborescence présentée dans la figure 3.12 (a)) au moins un qui est supérieur au seuil.
- ✓ Zéro tree (T) : indique que la valeur absolue du coefficient traité ainsi que tous ses descendants sont insignifiants par rapport au seuil considéré (inférieurs au seuil " t_0 ").

Tous les coefficients codés "*Positif*" ou "*Négatif*" seront mis dans une liste (en valeur absolue) selon l'ordre de parcours de ces coefficients, et leurs valeurs seront remplacées par zéro dans la matrice parcourue. Cette liste sera l'objet du traitement secondaire qu'on va détailler par la suite.

Après le parcours de toute la matrice, le seuil " t_0 " est divisé par deux, et l'opération est refaite (après avoir effectué le traitement secondaire) selon le nouveau seuil (t_1, t_2, \dots). Cette méthode est appelée "*quantification par approximations successives*" (SAQ : *Successive Approximation Quantization*) et peut être refaite tant que le seuil " t_i " est supérieur ou égal à 1, sachant qu'on a toujours la relation suivante :

$$\text{Equation 3.7} \quad t_i = (t_{i-1}) / 2$$

Traitement secondaire :

Si " t_i " est le seuil courant, alors les coefficients marqués (codés "*Positif*" ou "*Négatif*") dans la passe précédente (traitement principal) ont leurs valeurs absolues dans l'intervalle $[t_i, 2t_i]$. Cet intervalle est alors divisé en deux : $[t_i, 3t_i / 2]$ et $[3t_i / 2, 2t_i]$.

Les coefficients dont la valeur absolue se trouve dans le premier intervalle sont codés par le symbole "*0*", alors que ceux qui se situent (en valeur absolue) dans le second intervalle sont codés par le symbole "*1*". Lorsque le traitement secondaire est fini, l'algorithme reprend avec un nouveau seuil, soit " t_{i+1} ". Dans cette passe secondaire, un nouvel intervalle s'ajoute aux deux précédents : $[t_{i+1}, t_i]$. Ces trois intervalles seront alors raffinés au cours des itérations suivantes. Tout le processus de l'algorithme EZW est illustré par la figure suivante :

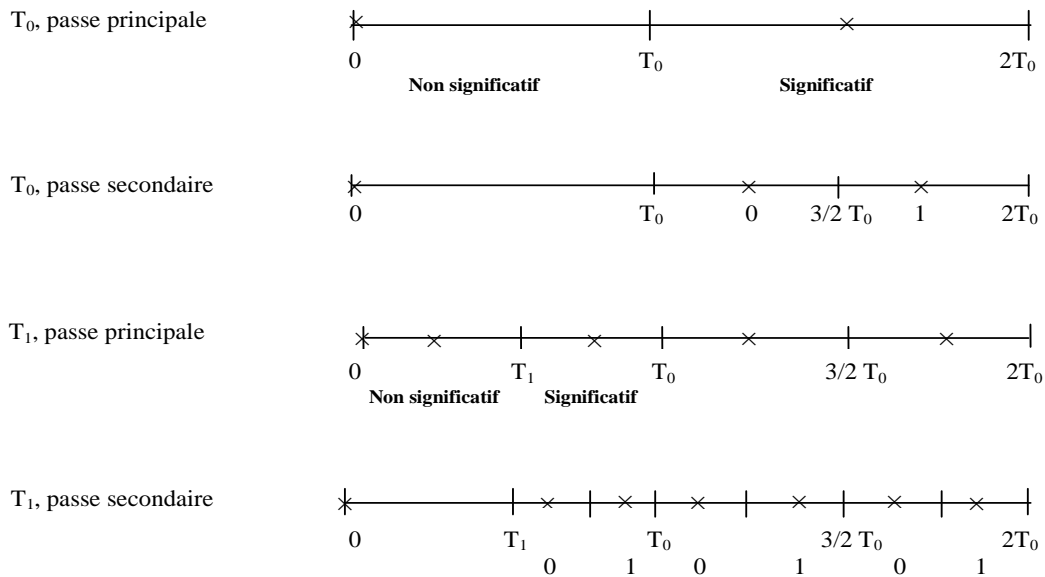


Figure 3.13. Déroulement des deux étapes de l'algorithme EZW pour deux seuils

- **La technique de codage**

L'algorithme *EZW*, appliqué à des coefficients d'ondelettes quantifiés, génère un code incluant plusieurs suites de "zéro tree" et de "zéro isolé". Cette caractéristique permet de favoriser l'utilisation du codage *RLE*. La chaîne de compression adoptée est représentée par la figure suivante :

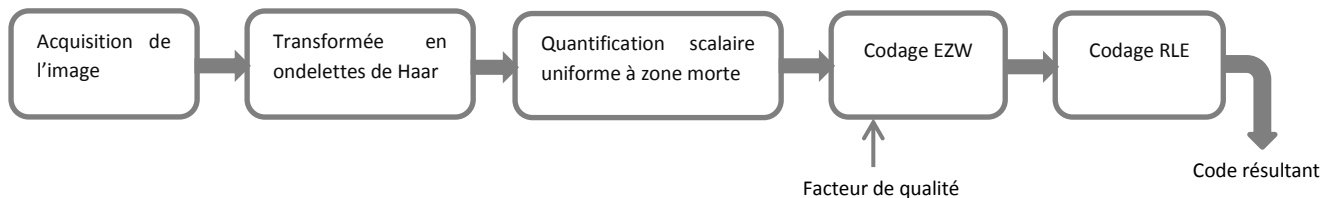


Figure 3.14 : Chaîne de compression d'image à base de la transformée en ondelettes de Haar, le codage EZW et le codage RLE

La figure 3.15 représente la caractéristique "débit-distorsion" du codeur implémenté. Le nombre d'itérations de l'algorithme *EZW* influe considérablement sur le taux de compression ainsi que la qualité de l'image reconstruite (*PSNR*). L'exécution de 8 itérations (nombre maximum d'itérations pour une image monochrome codée sur 8 bit/pixel) fournit une très bonne qualité d'image au prix d'un mauvais taux de compression (débit binaire de 4.5 bit par pixel (bpp)) et un nombre d'instructions assez élevé pour les nœuds de *RCSFs*. La figure 3.15 montre que la configuration du codeur *EZW* sur 5 ou 6 itérations répond le mieux au compromis "qualité de l'image reconstruite – taux de compression" ; le *PSNR* est aux alentours de 30 dB et le débit binaire varie entre 2.5 bpp et 3.5 bpp.

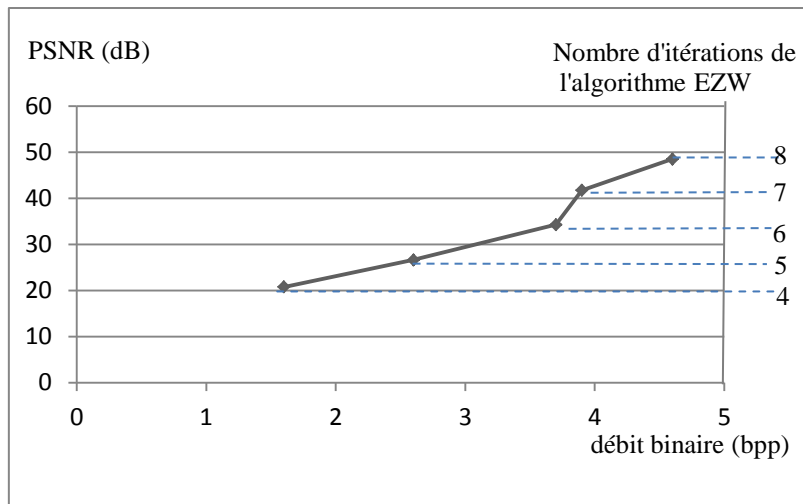


Figure 3.15 : Caractéristique "débit-distorsion" du codeur étudié

Afin de justifier nos choix, le codeur de la figure 3.14 a été implémenté sur un nœud de réseau de capteurs sans fil de type *MICA2*. L'objectif est de prouver l'intérêt d'une telle implémentation en termes de consommation énergétique par rapport au transfert d'images entières à travers le réseau. Les résultats obtenus sont présentés sur la figure suivante :

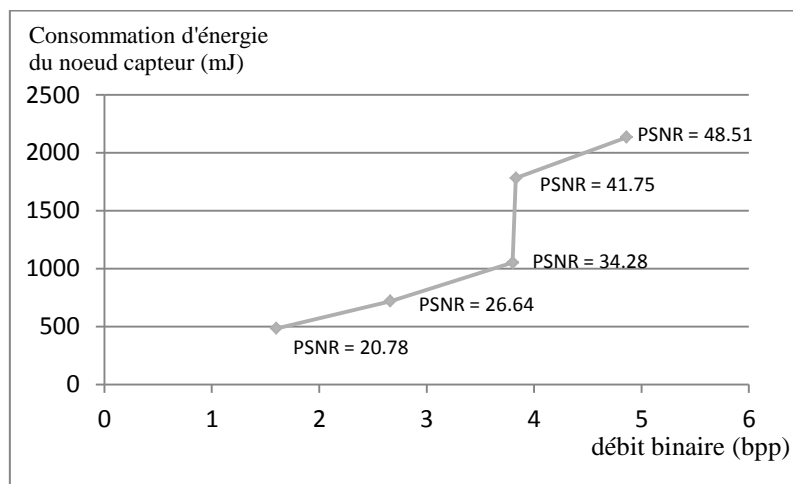


Figure 3.16 : Consommation énergétique d'un nœud de RCSF de type MICA2 lors de la compression d'une image (128x128 pixels – 8 bpp) et la transmission du code résultant

La figure 3.16 montre que la consommation du nœud augmente avec la qualité de l'image reconstruite. L'optimisation de l'énergie dissipée par le nœud nécessite un taux de compression relativement élevé, ce qui induit une mauvaise qualité d'image (20.78 dB pour un débit binaire de 1.6 bpp). Le meilleur compromis est atteint pour une consommation énergétique variant de 700 mJ à 1030 mJ ; autrement dit un PSNR qui varie de 26.6 dB à 34.28 dB et un débit binaire variant de 2.6 bpp à 3.8 bpp. A l'intérieur de cet intervalle, la consommation totale du nœud (traitement + transmission) est toujours inférieure à celle

obtenue lors de la transmission d'une image entière ayant les mêmes caractéristiques (le nœud consomme environ 2830 mJ pour transmettre l'image avec une puissance de transmission de 5dBm).

Certes les gains énergétiques apportés par le codeur implémenté sont importants, mais le taux de compression atteint par la chaîne de compression reste relativement faible. Un taux de compression élevé nécessite forcément un codeur entropique plus puissant. Le codage arithmétique permet, à partir des probabilités d'apparition des symboles à coder, d'atteindre des taux de compression élevés. Le problème majeur de cette technique est la complexité et le temps de calcul relativement élevés. En effet, le codeur doit calculer les probabilités d'apparition des symboles à coder avant de procéder au processus de codage. Chaque symbole est codé sur un nombre de bits inversement proportionnel à sa probabilité d'apparition dans la séquence à traiter par le codeur. Le calcul des probabilités se fait soit de façon statique (avant de commencer le codage proprement dit) ou dynamique (au fur et à mesure du codage des symboles). Le coût des deux approches est pratiquement le même ; dans les deux cas le calcul des probabilités est très coûteux en termes de complexité et temps de calcul.

Le codeur *RLE* de la figure 3.14 a été remplacé par un codeur arithmétique standard (non adaptatif). Il s'agit du même codeur intégré au sein de la chaîne de compression à base de la *DCT de Loeffler* (figure 3.5) ; mais le nombre d'instructions et la mémoire occupée pour effectuer le codage arithmétique ne sont plus les mêmes. En effet, le codeur traite des blocs de taille 8x8 ou 16x16 pixels dans le cas où il code une image transformée en appliquant la *DCT*. Dans ce cas-là, le codeur arithmétique nécessite des ressources énergétiques relativement élevées pour traiter indépendamment tous les blocs de l'image transformée par *DCT*. L'énergie consommée par ce même codeur est un peu plus élevée dans le cas d'une éventuelle implémentation avec la transformée en ondelettes (figure 3.17). Ceci est dû au fait de coder toute l'image transformée au lieu de la coder bloc par bloc. Le traitement, lié au codage arithmétique, nécessite donc des capacités de calcul considérables pour définir le modèle probabiliste des coefficients de l'image transformée, et les coder (les coefficients) par la suite.

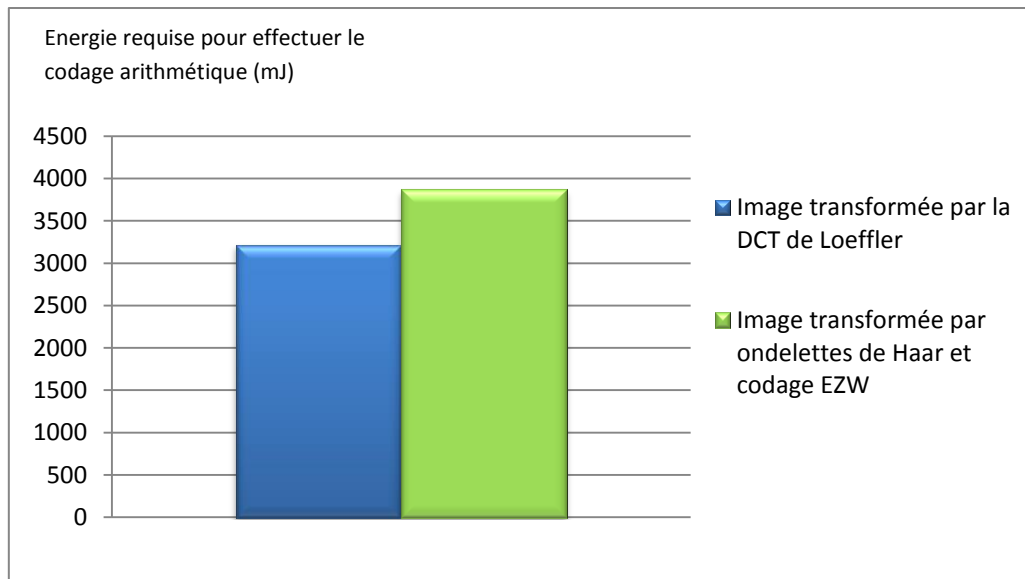


Figure 3.17 : Consommation énergétique d'un nœud de RCSF de type MICA2 lors du codage (en utilisant le codage arithmétique) d'une image transformée (128x128 pixels – 8 bpp)

Afin d'alléger le traitement lié au codage arithmétique, nous proposons une modification légère et efficace de l'algorithme standard. La modification consiste à utiliser une table statistique incluant les probabilités d'apparition des coefficients à coder. Cette table est construite en utilisant une image de référence ayant les mêmes caractéristiques que l'image à coder (images monochromes, de même taille, et ayant des pixels codés sur le même nombre de bits). En effet, nous avons constaté que le fait de changer une image par une autre ayant des caractéristiques semblables, aboutit à un faible changement des fréquences d'apparition des symboles à coder. Ce petit changement n'affecte pas trop les performances du codeur en termes de taux de compression et qualité de l'image reconstituée. Malgré que l'approche proposée n'est basée que sur des observations et des résultats expérimentaux, plusieurs couples "*Image de référence – Image à coder*" ont été utilisés afin de la valider.

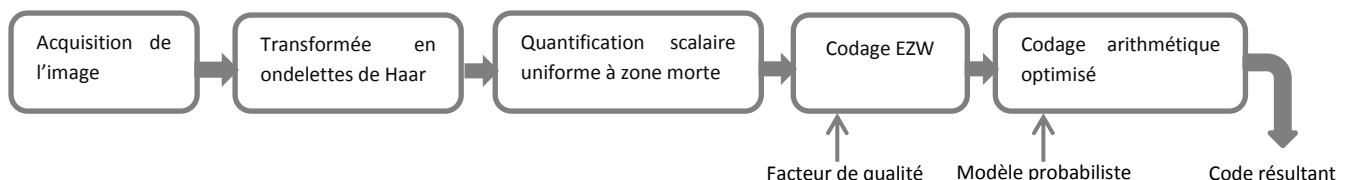


Figure 3.18 : Chaîne de compression d'image à base de la transformée en ondelettes de Haar, le codage EZW et une version améliorée du codage arithmétique

Le codeur arithmétique de l'image 3.18 code des images transformées, quantifiées et pré-codées (en utilisant le codage EZW). La transformée en ondelettes et l'étape de quantification génèrent toujours un nombre limité de coefficients significatifs (coefficients de grandes valeurs représentant les basses fréquences) et un grand nombre de coefficients ayant

de faibles valeurs (coefficients non significatifs qui représentent les hautes fréquences). Cette caractéristique est toujours vraie quel que soit l'image à coder. Par conséquent, les fréquences d'apparition des coefficients de l'image transformée et quantifiée varient peu si on change une image par une autre. Le codage EZW permet de réorganiser les coefficients de l'image transformée (du plus significatif au moins significatif) ; il n'a pas donc d'influence sur le modèle probabiliste des coefficients à coder.

Par contre, le modèle probabiliste adopté change en fonction des caractéristiques de l'image à coder (taille, nombre de bits par pixel, etc.) ainsi que le niveau de transformation appliqué (la transformée en ondelettes est appliquée une seule fois, deux fois,...). Le codeur implémenté devrait être capable de s'adapter avec les besoins de l'application en chargeant un modèle par un autre de façon dynamique s'il le faut. Mais généralement, dans une application de réseaux de capteurs de vision sans fil, les caractéristiques de l'image et le niveau de transformation sont des paramètres prédéfinis à l'avance. L'utilisateur fixe dès le départ ces paramètres en fonction de la qualité de service (QoS) souhaitée. Par conséquent, un seul modèle statistique devrait être utilisé pour chaque application (en fonction des besoins de l'utilisateur).

La figure suivante (figure 3.19) présente les performances de la chaîne de compression proposée (figure 3.18). Les deux images de référence "*Lena*" et "*Cameraman*" ont été utilisées afin de dégager la caractéristique "*Débit-Distorsion*" du codeur. La compression et le codage ont été appliqués sur l'image "*Cameraman*" en utilisant le modèle probabiliste de l'image "*Lena*".

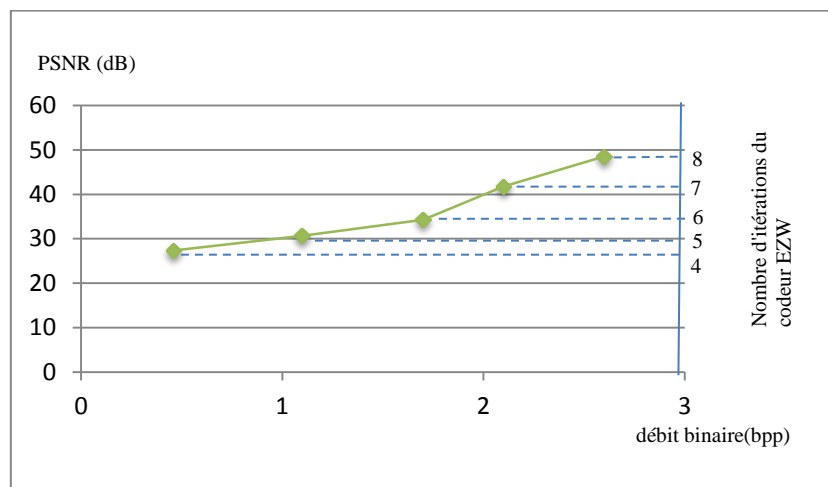


Figure 3.19 : Caractéristique "débit-distorsion" du codeur étudié

La caractéristique "*Débit-Distorsion*" du codeur adopté montre qu'il est plus performant que la chaîne de compression de la figure 3.14 (figures 3.15 et 3.19). A qualité d'image égale, le nouveau codeur proposé (à base du codage *arithmétique modifié*) atteint un taux de compression plus élevé que celui basé sur le codage *RLE*. (Un débit binaire de 1.5 bpp contre 3.4 bpp pour un PSNR de 32 dB). Ces résultats justifient la puissance du codage *arithmétique modifié* par rapport au codage *RLE*. Il reste à étudier le coût énergétique de cette approche afin de justifier l'adoption de la nouvelle chaîne de compression. L'implémentation du codeur de la figure 3.18 sur un nœud de réseau de capteurs sans fil (plateforme *Mica2*) a donné les résultats présentés sur la figure suivante :

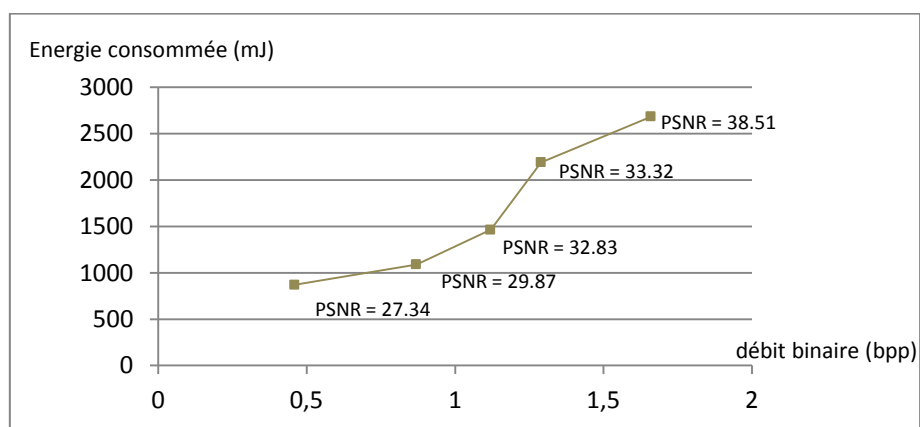


Figure 3.20 : énergie consommée par le nœud capteur (plateforme *Mica2*) lors de la compression (codeur de la figure 3.18) d'une image (128x128 pixels codés sur 8 bpp) puis la transmission du code résultant.

Vu que le *codage arithmétique* et le *codage RLE* sont deux codeurs sans pertes, l'implémentation de l'un ou de l'autre n'a pas une influence sur la qualité de l'image reconstruite. Par contre, la complexité des deux techniques de codage a une influence sur le taux de compression de l'image (bit rate) ainsi que l'énergie consommée par le nœud lors du codage. A consommation énergétique égale, la chaîne de compression à base du *codage RLE* fournit une meilleure qualité d'image (écart variant de 2 à 4 dB) mais un taux de compression moins bon (de 3 à 4.75 fois moins bon). La résolution de ce compromis nécessite de définir les objectifs fixés par l'utilisateur. L'optimisation de l'énergie dissipée par le nœud source du réseau (celui qui effectue la compression) privilégie l'utilisation de la première chaîne de compression (basée sur le codage *RLE*) puisqu'elle fournit une meilleure qualité d'image à consommation égale. Par contre, l'optimisation de l'énergie consommée par la totalité du réseau (l'ensemble des nœuds du réseau) favorise l'implémentation de la nouvelle approche (technique de compression à base du codage *arithmétique modifié*). En effet, le nœud source consomme un peu plus d'énergie pour fournir la même qualité d'image, mais un taux de

compression beaucoup plus élevé : un PSNR de 27 dB nécessite une consommation de 730 mJ pour la technique basée sur le *codage RLE* et 870 mJ pour la technique de compression à base du *codage arithmétique modifié*; en revanche, la chaîne à base du *codage RLE* atteint un débit binaire de 2.7 bpp contre 0.46 bpp pour la nouvelle chaîne de compression. Dans le cas de la nouvelle approche adoptée et à qualité d'image égale, l'information utile circulant sur le réseau représente le 1/5^{ème} par rapport aux données transmises dans le cas de l'approche basée sur le *codage RLE*. Sachant que le transcepteur radio (Émetteur/Récepteur radio) est le bloc le plus gourmand en énergie pour les nœuds de réseaux de capteurs sans fil, diviser par cinq le nombre de paquets circulant à travers le réseau est un atout intéressant pour la nouvelle approche.

En ce qui concerne la consommation énergétique, relativement élevée, du nœud source lors de l'implémentation de la nouvelle technique de compression, une implémentation matérielle du codeur paraît une solution efficace pour optimiser la puissance dissipée par ce nœud. Les détails de la solution matérielle seront discutés dans le chapitre suivant.

3.3. Approches de la compression vidéo pour les réseaux de capteurs de vision sans fil

La compression d'image permet de réduire la redondance spatiale des données d'une image. Ce type de compression est efficace dans le cas où on l'applique sur des séquences vidéo à faible cadence (1 image/seconde par exemple). Dans ce cas-là, la redondance temporelle entre les images est relativement faible ; l'implémentation de techniques de compression d'image est favorisée. Par contre, certaines applications nécessitent une moyenne ou forte cadence vidéo ce qui augmente la redondance temporelle entre les trames de la vidéo ; l'usage d'un codeur vidéo est privilégié dans ce cas-là. Un codeur vidéo efficace a pour objectif de réduire au maximum la redondance spatio-temporelle afin d'atteindre des taux de compression élevés. Optimiser le compromis "*efficacité – complexité*" d'un codeur vidéo est une tâche difficile à atteindre surtout dans le cas particulier des réseaux de capteurs de vision sans fil (figure 3.21), systèmes fortement contraints en ressources (mémoire, énergie et capacités de calcul).

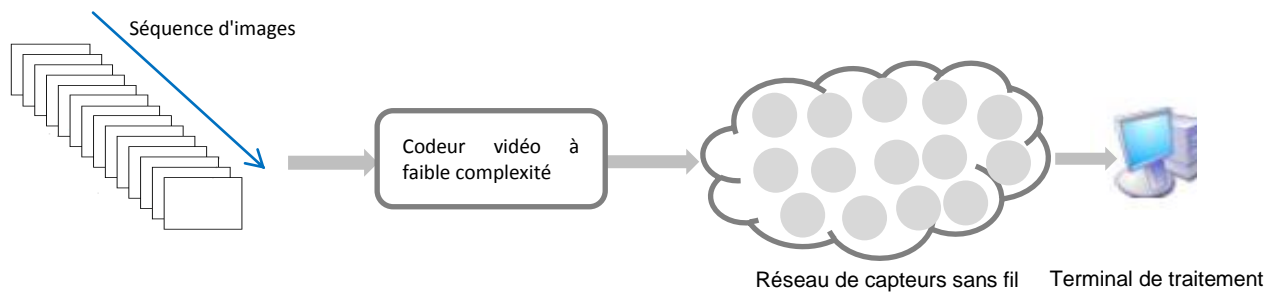


Figure 3.21 : Système de vision pour les réseaux de capteurs sans fil

Les codeurs vidéo classiques (MJPEG2000, MPEG-x, H.26x...) souffrent d'une forte complexité de calcul et sont loin de satisfaire les besoins spécifiques des réseaux de capteurs sans fil. La complexité algorithmique de ce type de codeurs s'explique par le fait d'appliquer plusieurs techniques à la fois sur le flux vidéo à coder :

- La technique de codage temporel différentiel qui part du constat que deux images successives I_n et I_{n+1} sont fortement semblables. Il est donc plus pratique de coder I_n puis $(I_n - I_{n-1})$ au lieu de coder les deux images séparément [98].
- La technique de compensation de mouvement qui part du constat qu'une séquence vidéo est souvent composée par des objets qui traversent le champ de la caméra. Ce "*travelling*" ou déplacement d'objets peut être modélisé par translation des blocs de l'image qui représentent ces objets en mouvement. Le codeur transmet les vecteurs "*translation*" au lieu de transmettre les blocs représentant les objets en mouvement. Parmi les techniques de compensation de mouvement, le "*Block Matching*" est la technique la plus connue. La présente technique permet de rechercher pour un bloc de l'image I_{n+1} un bloc de l'image I_n qui soit le plus proche. Le critère d'appariement des blocs utilisé est l'erreur quadratique moyenne [98].
- Techniques de compression d'image classiques (JPEG, JPEG2000...). Ces techniques sont utilisées pour réduire la redondance spatiale de certaines images de la séquence vidéo. Les techniques de réduction des redondances temporelles seront appliquées aux autres images de la séquence [98].

Les techniques de compression vidéo classiques sont généralement appliquées à des séquences vidéo rapides et nécessitant une forte cadence d'images. Par contre, les réseaux de capteurs sans fil actuels sont loin d'être utilisés pour traiter et transmettre de la vidéo à haute cadence. Il s'agit de réseaux très bas débit taillés pour des applications spécifiques. Faire de la vidéo-surveillance, surveiller un parc naturel, ou encore détecter un incendie, sont des

applications multimédias qui font appel à des séquences vidéo à moyenne voire faible cadence d'images. En plus, ce genre d'applications favorise l'emploi de plusieurs caméras fixes sous différents angles d'observation que d'utiliser une seule caméra mobile. Les séquences enregistrées dans ce cas-là contiennent beaucoup de redondance temporelle vu la nature des applications visées et la non-mobilité de l'imageur.

Tous ces paramètres nous ont inspirés pour développer une nouvelle technique de compression vidéo (figure 3.22), à la fois efficace et à faible complexité. L'algorithme proposé exploite la redondance spatio-temporelle ; il permet de détecter les blocs qui changent significativement au cours du temps (images I_n et I_{n-1}). Pour chaque bloc de l'image courante, on calcule la moyenne des pixels de ce bloc et on compare cette moyenne à celle du même bloc de l'image précédente. L'utilisateur définit à l'avance un seuil S_0 tel que :

*Si (Moyenne des pixels du bloc B_i de l'image I_n - Moyenne des pixels du bloc B_i de l'image I_{n-1}) $> S_0$:
changement significatif du bloc B_i ; compression du bloc B_i de l'image I_n puis transmission du code résultant ;
passer au bloc suivant : B_{i+1}*

*Si (Moyenne des pixels du bloc B_i de l'image I_n - Moyenne des pixels du bloc B_i de l'image I_{n-1}) $< S_0$:
changement non significatif du bloc B_i ; passer au bloc suivant : B_{i+1}*

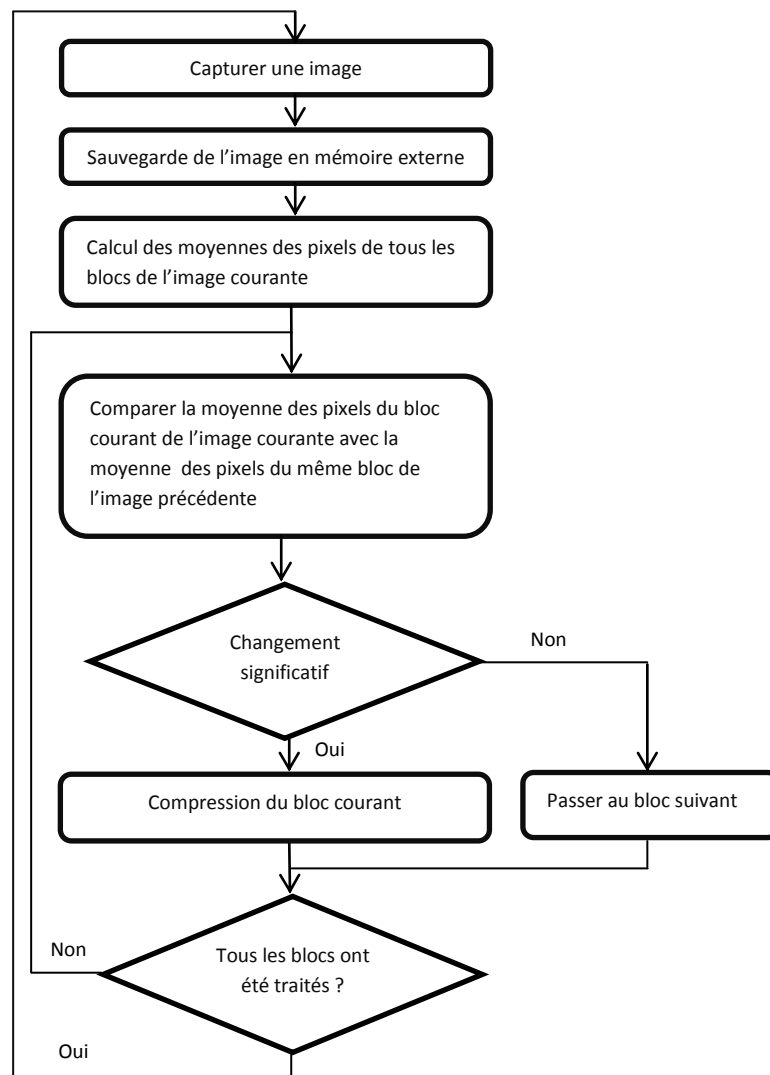


Figure 3.22 : L'algorithme de compression vidéo proposé

La chaîne de compression de la figure 3.14 (HWT-Quantification-EZW-RLE) a été employée pour coder les blocs désignés de l'image courante (les blocs dont la moyenne a changé significativement entre deux trames successives). Ce choix a été fait pour deux raisons :

- L'utilisation du codage arithmétique standard est très coûteuse en termes de complexité et temps de calcul.
- L'implémentation du codage arithmétique modifié ne peut pas être appliquée à des séquences vidéo. Chaque séquence vidéo possède ses propres caractéristiques (nombre d'images, nombre d'objets en mouvement, vitesse de mouvement des objets, nombre et position des blocs à coder pour chaque image de la séquence, etc.). En plus, le codage arithmétique modifié nécessite de stocker en mémoire et pour chaque bloc de

l'image à coder le modèle statistique adéquat, ce qui n'est pas pratique dans le cas particulier des réseaux de capteurs sans fil.

Pour un codeur vidéo, l'enjeu principal est d'atteindre un taux de compression élevé tout en gardant une bonne qualité d'image et un nombre d'instructions acceptable. Le tableau suivant justifie le choix que nous avons adopté. Les séquences de test utilisées sont des vidéos composées par des images de taille 176x144 pixels (8 bpp).

	Énergie totale consommée (mJ/trame) : traitement + transmission			
Codeur →	H.264 Inter[99]	H.264 Intra[99]	MJPEG2000 Intra[99]	Notre codeur
Séquence de test ↓				
Mobile	769.86	64.64	20.39	23.94
Carphone	757.88	57.71	11.42	13.78
Foreman	763.32	57.74	13.65	14.23

Tableau 3.1 : Énergie consommée par différents codeurs vidéo implémentés sur la plateforme STARGATE (plateforme de réseaux de capteurs sans fil)

Notre codeur vidéo consomme beaucoup moins d'énergie que le H.264. Même le bloc le moins complexe du codeur H.264 (H.264 intra-trame) dissipe plus d'énergie que le nôtre. Les consommations de MJPEG2000 intra-trame et de notre codeur sont très proches pour les trois séquences de test ; ceci montre que la complexité algorithmique de la technique que nous avons proposée est comparable à celle de MJPEG2000 intra-trame. Les auteurs des travaux de recherche de la Réf [99] n'ont pas implémenté MJPEG2000 inter-trame ; la consommation totale du codeur MJPEG2000 (intra et inter-trame) excèderait certainement l'énergie dissipée par notre codeur.

Certes, des codeurs vidéo standards comme le H.264, MJPEG2000 ou encore MPEG sont plus performants que notre codeur en termes de caractéristique "*débit-distorsion*", mais ces codeurs occupent trop de ressources (mémoire, énergie, temps de calcul, etc.) et sont loin d'être implémentés sur des nœuds de réseaux de capteurs sans fil. Les deux figures suivantes présentent la caractéristique "*débit-distorsion*" du codeur étudié. La séquence vidéo utilisée pour extraire cette caractéristique représente l'exemple typique d'applications multimédias dédiées aux réseaux de capteurs sans fil : il s'agit d'une séquence vidéo monochrome et de petite taille (320x240 pixels codés sur 8 bpp) ; la caméra bouge lentement et filme le passage de plusieurs individus devant la caméra. Ce type de vidéo répond bien aux contraintes spécifiques des réseaux de capteurs sans fil ; on est encore loin d'utiliser des vidéos rapides et Haute- Définition pour ce type de réseaux.

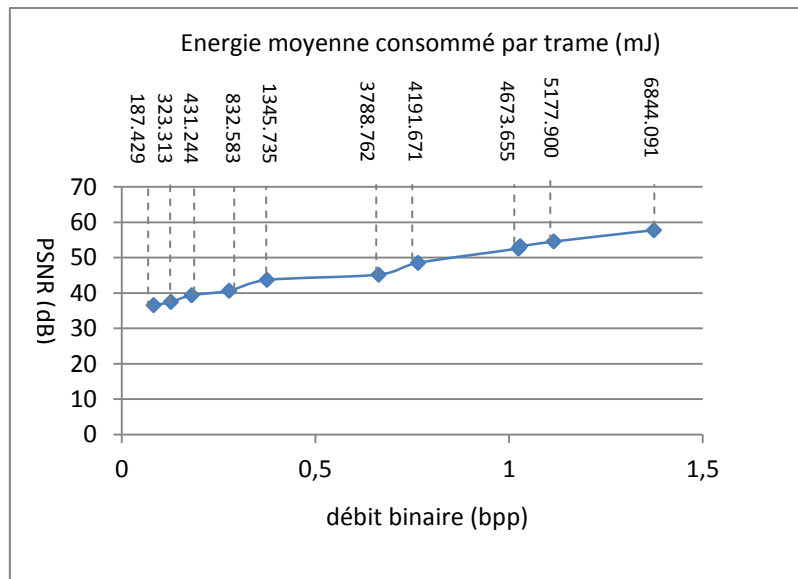


Figure 3.23 : Caractéristique débit-distorsion du codeur (blocs de taille 8x8 pixels – 8bpp) et consommations typiques du nœud (plateforme MICA2) lors de l’implémentation du codeur

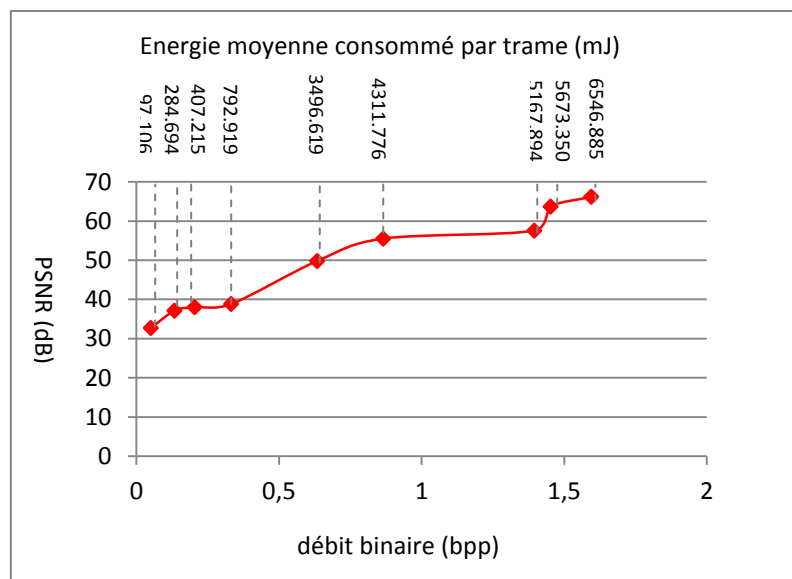


Figure 3.24 : Caractéristique débit-distorsion du codeur (blocs de taille 16x16 pixels – 8bpp) et consommations typiques du nœud (plateforme MICA2) lors de l’implémentation du codeur

Les figures 3.23 et 3.24 montrent que l’énergie consommée par le nœud source du réseau de capteurs sans fil (celui qui applique le codage) varie en fonction du taux de compression cherché (et par conséquent la qualité de la vidéo souhaitée). Même dans le cas où le codeur atteint des taux de compression élevés, le PSNR de la vidéo reste acceptable (>30dB). Dans ce cas-là, les gains énergétiques sont très intéressants par rapport à la transmission de vidéos entières sans compression (jusqu’à 97% pour la figure 3.23 et jusqu’à 99% pour la figure 3.24).

Malgré les gains énergétiques importants atteints lors de l'implémentation logicielle du codeur vidéo proposé, l'ordre de grandeur des consommations ne satisfait pas les attentes des utilisateurs de réseaux de capteurs sans fil. La complexité de tous les codeurs implémentés dans ce chapitre dépasse largement les capacités des nœuds capteurs. L'implémentation matérielle de ces techniques de compression (d'images et de vidéos) permettrait d'optimiser au maximum l'énergie consommée par le nœud source du réseau.

3.4. Conclusion

Dans ce chapitre, nous avons justifié l'intérêt de la compression dans le cas particulier des applications liées au transfert d'images à travers les réseaux de capteurs sans fil. Les implémentations logicielles des techniques de codage proposées ont montré leur efficacité en termes de gains énergétiques apportées à la source et sur la totalité du réseau. Ces gains peuvent être améliorés davantage par une implémentation matérielle des codeurs présentés. Le chapitre suivant détaille l'implémentation matérielle de tout le système de vision (capteur d'image CMOS et codeurs) et justifie les choix architecturaux adoptés.

Chapitre 4

De l'implémentation matérielle du système de vision vers l'implémentation du réseau de capteurs de vision sans fil idéal

Introduction

Ce chapitre sera réservé à l'implémentation matérielle du système de vision adopté, système intégrant des fonctionnalités de compression d'image et de vidéo. Les performances des codeurs sélectionnés ont été présentées dans le chapitre précédent. Dans cette partie, nous nous intéressons à l'étude des architectures adoptées et les performances des implémentations matérielles [90, 91]. Nous montrerons aussi l'intérêt de l'intégration de la compression d'image sur le plan focal de l'imageur, par rapport aux implémentations logicielles de ces techniques de compression.

4.1. Architecture du système de vision

Le système de vision adopté englobe le capteur d'image CMOS, le convertisseur Analogique/Numérique et le codeur vidéo. Ces différents blocs vont être détaillés dans les sections suivantes.

4.1.1. Le capteur d'image CMOS

L'utilisation d'un capteur d'image CMOS numérique *DPS* (*Digital Pixel Sensor*) n'est pas appréciée dans le cas de réseaux de capteurs sans fil. En effet, un DPS implique l'utilisation d'un nombre élevé de transistors par pixel (allant de quelques dizaines à quelques centaines de transistors). Un pixel numérique intègre généralement de la mémoire, un Processeur Élémentaire (PE), et un Convertisseur Analogique/Numérique (CAN). Ce type d'imageurs consomme trop d'énergie puisque chacune des opérations d'acquisition de l'image, de conversion Analogique/Numérique, de stockage des informations, ainsi que du prétraitement de ces informations, s'effectue simultanément au niveau de tous les pixels, d'où un fort appel de courant [31]. Les imageurs *DPS* permettent d'atteindre des vitesses

d'acquisitions allant jusqu'à 10.000 images/seconde. Ils sont généralement employés dans des applications qui n'ont pas des contraintes énergétiques et de surface strictes.

Notre choix s'est fixé sur une matrice de pixels actifs à trois transistors. L'APS-3T (pixel actif à 3 transistors: APS - 3T) est utilisé par un grand nombre d'applications et de systèmes nécessitant une économie importante d'énergie (Webcams, Smartphones, etc.) [31]. La figure 4.1 présente un exemple de *Layout* du pixel standard (APS - 3T), réalisé au laboratoire TIMA en technologie CMOS 0.35 μm d'AMS.

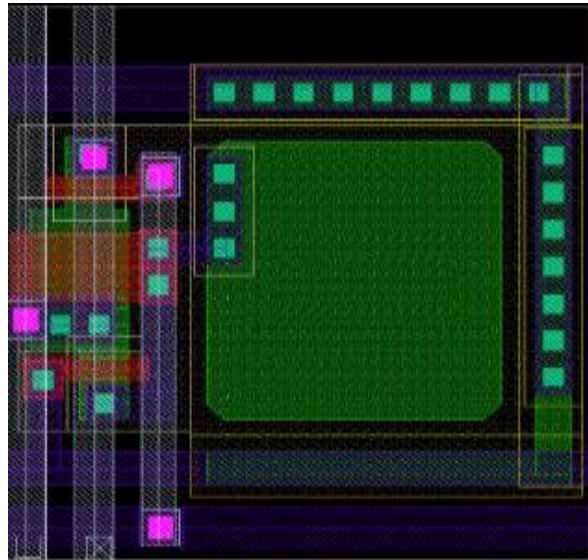


Figure 4.1 : Layout d'un pixel standard

L'architecture globale de l'imageur adopté est la suivante (figure 4.2) :

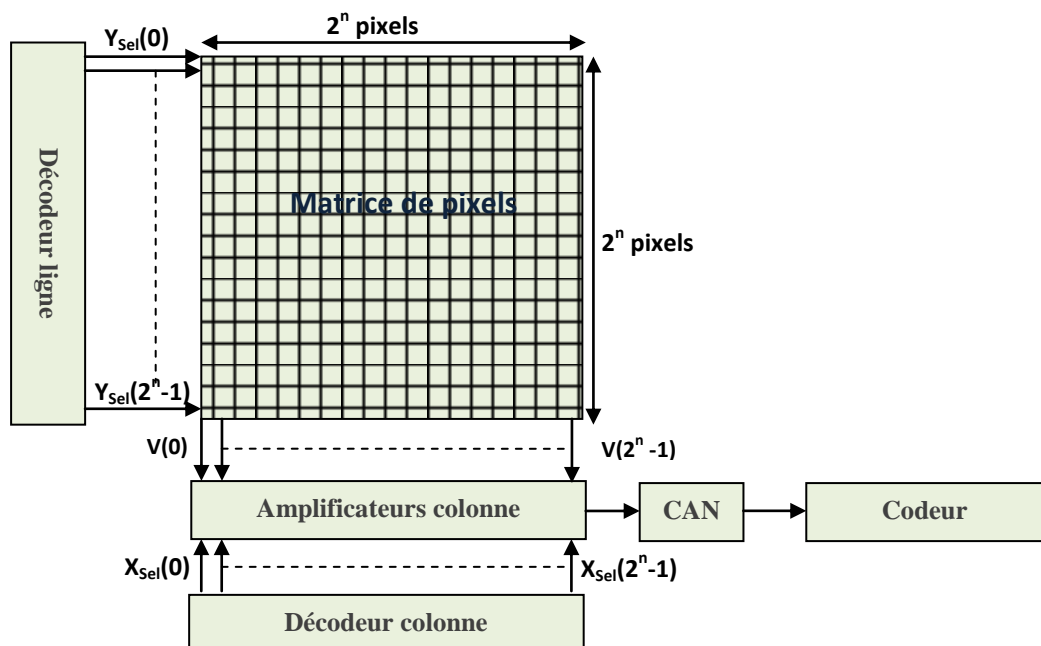


Figure 4.2 : Architecture globale du système de vision adoptée

L'implémentation d'un codeur vidéo par colonne engendre un coût supplémentaire considérable (énergie consommée et surface occupée par les codeurs). Rappelons que l'objectif principal de l'étude est l'intégration du système de vision conçu au sein d'un nœud de réseau de capteurs sans fil. Ces nœuds fonctionnent généralement à des fréquences basses (quelques MHz) ; pour ces hôtes, l'économie d'énergie et le coût de fabrication sont plus prioritaires que le temps de traitement. Pour ces raisons, nous avons opté pour cette architecture qui inclut un seul CAN et un seul codeur vidéo. Le Convertisseur Analogique/Numérique ne devrait pas être très rapide afin d'optimiser sa consommation. Cette condition est satisfaite lorsqu'on traite (conversion Analogique/Numérique) quelques images par seconde (< 10 image/seconde), de taille inférieure ou égale à 352×288 pixels [31]. Ceci permet de mieux répondre aux exigences particulières des réseaux de capteurs de vision sans fil (On ne cherche pas à transmettre des images de haute résolution à travers les RCSFs. On ne cherche non plus à transmettre un nombre important d'images par seconde à travers ce type de réseaux).

Les décodeurs ligne et colonne ont été décrits en VHDL. Le décodeur ligne, à partir d'une adresse $Y_{sel}(i)$, connecte l'ensemble des pixels de la ligne sélectionnée aux bus de lecture colonne de la matrice. Par la suite, et à partir d'une adresse $X_{sel}(j)$, le décodeur colonne connecte le bus colonne sélectionné à la sortie.

Au pied de la colonne de la matrice, il y a un amplificateur par colonne de pixels (d'où l'appellation *amplificateur colonne*). Les amplificateurs colonne reçoivent simultanément les signaux en provenance de la ligne active de pixels. Les amplificateurs colonne seront par la suite parcourus séquentiellement afin de fournir les signaux amplifiés au CAN. La figure 4.3 présente l'architecture de l'ensemble formé par le pixel standard et l'amplificateur colonne standard utilisés.

La figure 4.3, présentée ci-dessous, détaille l'architecture du pixel standard et de l'amplificateur colonne standard:

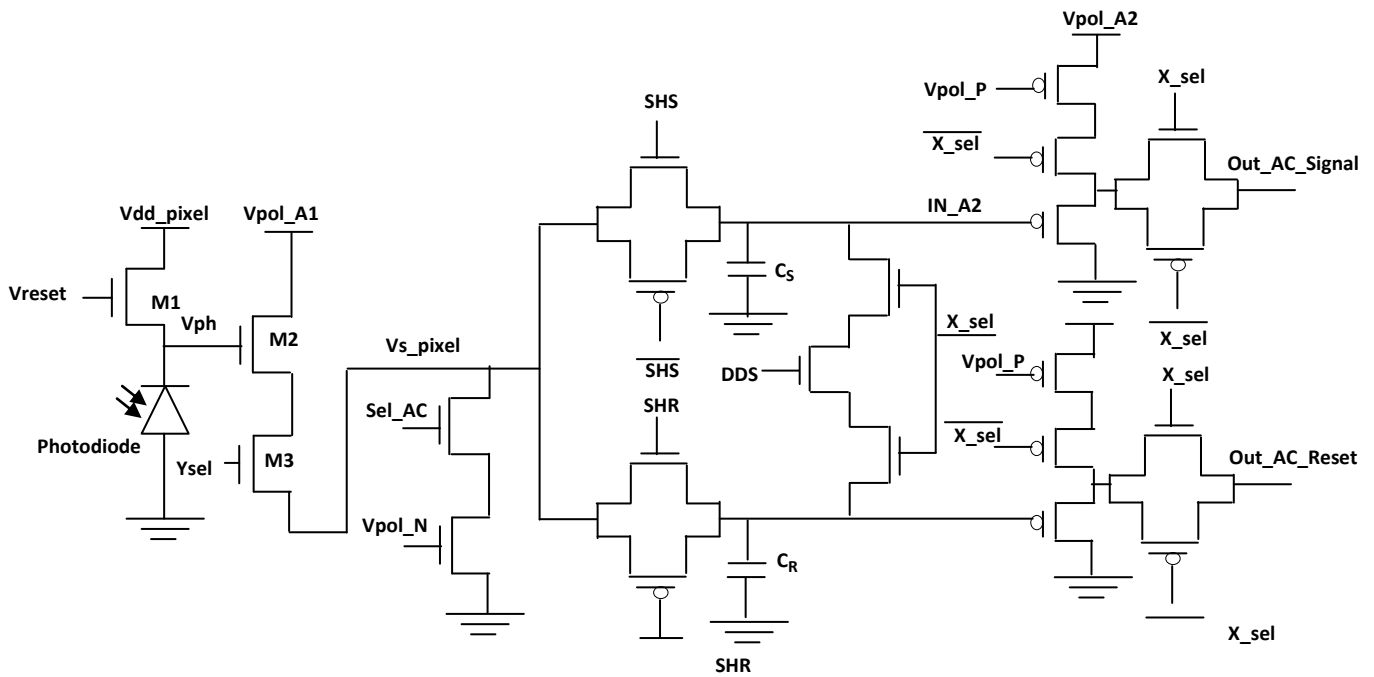


Figure 4.3 : Pixel standard et amplificateur colonne

4.1.2. Conversion Analogique/Numérique

En ce qui concerne la conversion Analogique/Numérique, un CAN de commerce (CAN à approximations successives) a été utilisé par des anciens doctorants de notre équipe de recherche afin de convertir les signaux en provenance de la matrice de pixels. L'étude détaillée du convertisseur Analogique/Numérique ne fait pas partie des objectifs de cette thèse.

4.1.3. Codeurs pour la compression d'image

Le partitionnement Analogique/Numérique est une étape primordiale à préparer lors de la conception de nos codeurs. Dans la littérature, seules les transformées en ondelettes et l'étape de quantification peuvent être conçues à la fois en analogique et en numérique. Le reste des techniques de codage étudiées (*EZW*, *RLE* et *codage arithmétique*) n'ont jamais été implémentées en analogique vue la complexité algorithmique considérable de ces algorithmes. Il reste donc à confirmer s'il est plus intéressant d'implémenter la transformée en ondelettes de Haar et la quantification en analogique ou en numérique.

On a choisi de commencer par l'étude de l'approche analogique qui consiste à concevoir un bloc analogique au pied de la colonne de l'imageur, permettant d'appliquer la transformée en ondelettes de Haar. Le bloc intègre des capacités commutées qui forment un système capable de calculer à la fois la moyenne et la moyenne de la différence de deux pixels

adjacents. L'idée proposée est inspirée des travaux de Tristan Thabuis, un ancien doctorant de notre équipe de recherche, qui a essayé d'implémenter une version modifiée de la transformée en ondelettes discrète au pied de la colonne de l'imageur [50]. Malheureusement, la solution proposée dans [50] ne permet pas d'avoir les mêmes résultats que ceux obtenus lors de l'implémentation de la transformée en ondelettes discrète standard. La version modifiée ne permet pas de coder de manière efficace (en utilisant les techniques de codage étudiées précédemment) les coefficients d'ondelettes générés.

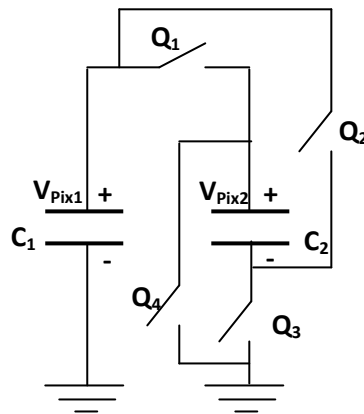


Figure 4.4 : Schéma simplifié de la circuiterie de calcul à base de capacités commutées

Plusieurs contraintes majeures sont apparues lors des tests effectués. Les résultats de simulation sous l'environnement *Cadence* ont montré que l'approche analogique manque de précision. Le calcul des coefficients d'ondelettes n'est pas précis à cause de plusieurs paramètres dont les capacités parasites et le nombre important d'ouvertures et de fermetures des interrupteurs commandés. Les erreurs de précision existent déjà avant même de procéder au dessin des masques, étape qui nécessite un bon appairage des capacités et l'utilisation d'une technique de dessin symétrique, ce qui n'est pas facile à réaliser. Un mauvais appairage et une mauvaise symétrie pourraient dégrader davantage la précision des coefficients calculés.

D'autres contraintes nous ont poussés à choisir l'approche numérique pour calculer les coefficients d'ondelettes. En effet, les coefficients calculés lors du traitement des lignes de l'image devraient être sauvegardés puisqu'ils serviraient par la suite pour calculer la transformée en ondelettes sur les colonnes. La façon la plus pratique consiste à sauvegarder les coefficients calculés au sein des pixels (en écrasant les anciennes valeurs) ce qui nécessite une circuiterie supplémentaire. Le même traitement (calcul de la transformée en ondelettes de

Haar) sera exécuté de nouveau, mais cette fois-ci sur les colonnes de l'image. La lecture des pixels devrait être faite colonne par colonne, ce qui n'est pas facile à gérer.

Dans la littérature, et pour des raisons de simplicité des calculs, on parle toujours d'implémentation analogique de "*la transformée en ondelettes modifiée*" au lieu de la "*transformée en ondelettes classique*". La modification consiste à traiter chaque bloc de l'image, de taille 2x2 pixels, à part entière [50, 81]. Le but de la modification est de simplifier le traitement et par conséquent simplifier l'implémentation analogique de la transformée. Quatre coefficients sont calculés pour chaque bloc, mais ces coefficients ne représentent pas la transformée en ondelettes réelle et ils ne peuvent pas être exploités efficacement par le codage EZW.

Les codeurs étudiés au sein du troisième chapitre seront donc implémentés entièrement en numérique. Dans la partie suivante, l'implémentation matérielle et l'étude de performances de ces codeurs feront l'objet d'une étude détaillée.

A. Codeur à base de la transformée en ondelettes de Haar, le codage EZW et le codage RLE

Le schéma bloc du codeur implémenté est présenté par la figure suivante :

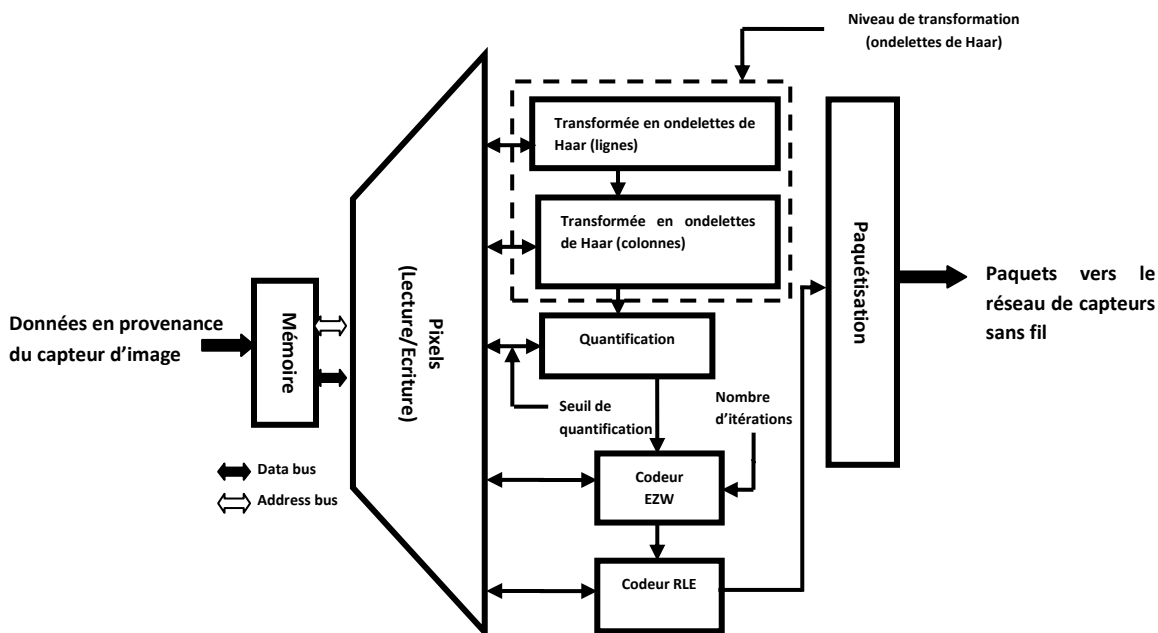


Figure 4.5 : Schéma bloc du codeur étudié (DWT-Quantification-EZW-RLE)

L'outil ISE 9.1 de Xilinx a été utilisé afin d'extraire les résultats de synthèse de l'architecture proposée. Le tableau suivant illustre ces résultats pour une implémentation sur une carte FPGA Xilinx Virtex 5. Le codage a été appliqué sur une image de taille 128x128 – 8bpp.

	Fréquence de fonctionnement (MHz)	Temps de traitement (ms)	Puissance dynamique consommée (mW)	Puissance statique dissipée (mW)	Energie consommée (mJ)
Codeur étudié	139.49	4.10	267	4	1.09

Tableau 4.1 : Performances du codeur étudié : implémentation sur une carte FPGA Xilinx Virtex 5

Le tableau 4.1 montre que la puissance dynamique est largement supérieure à la puissance statique lors de l'implémentation matérielle du codeur présenté. Le circuit, implémenté sur carte FPGA, consomme environ 1 mJ. Par contre, le nœud de réseau de capteurs sans fil (plateforme MICA2) consomme entre 280 mJ et 1090 mJ pour coder l'image (sans compter le coût énergétique de la transmission du code résultant par le nœud source à travers le réseau). Le coût du traitement lié au codage varie en fonction du nombre d'itérations du codage EZW, le nombre de niveaux appliqués pour la transformée en ondelettes de Haar, et le seuil de quantification appliqué. Nous constatons ici que l'implémentation matérielle a permis d'atteindre des gains énergétiques considérables et précieux pour le nœud source du réseau de capteurs sans fil. Ces gains ainsi que les performances globales du circuit peuvent être améliorés davantage dans le cas d'une éventuelle implémentation sur ASIC (Application-Specific Integrated Circuit). Le tableau suivant présente les résultats de synthèse obtenus dans ce cas-là.

	Technologie utilisée	F max (MHz)	Puissance dynamique (mW)	Surface du codeur (mm ²)
Codeur étudié	45 nm	840	17.89	0.061
	90 nm	436	19.78	0.15
	0.13 µm	383	26.66	0.2
	0.18 µm	319	38.91	0.54
	0.35 µm	187	53.35	2.19
	0.6 µm	115	71.12	9.85

Tableau 4.2 : Performances du codeur étudié : implémentations ASIC pour différentes technologies standards d'intégration

L'outil *Design Vision Tool* de Synopsys a été utilisé pour la synthèse logique de l'architecture proposée. L'outil *SOC Encounter* de Cadence a été utilisé pour le placement et le routage. Les résultats illustrés ci-dessus (tableau 4.2) montrent que l'implémentation sur

ASIC du codeur est très bénéfique en termes de fréquence de traitement, puissance dynamique consommée et surface occupée par le codeur. L'implémentation ASIC du codeur permet d'atteindre des vitesses de calcul plus élevées, une puissance dynamique dissipée beaucoup moins importante, et une surface occupée plus petite, en comparaison avec les résultats obtenus lors de l'implémentation du même codeur sur une carte FPGA Xilinx-Virtex-5.

B. Codeur à base de la transformée en ondelettes de Haar, le codage EZW et le codage arithmétique modifié

Cette chaîne de compression est plus performante en comparaison avec le premier codeur étudié. Le codage *RLE* a été remplacé par le *codage arithmétique modifié*, technique de compression plus performante mais aussi plus complexe que la technique *RLE*. Le schéma bloc du codeur implémenté est illustré par la figure suivante :

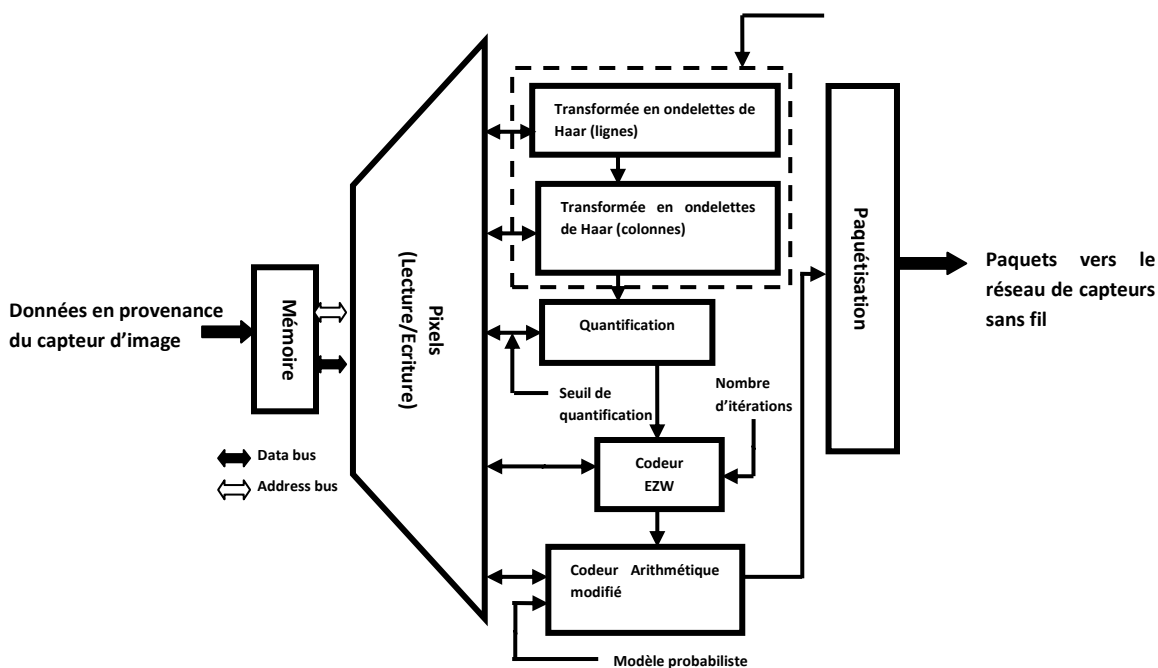


Figure 4.6 : Schéma bloc du codeur étudié (DWT-Quantification-EZW-Arithmétique modifié)

Les résultats de synthèse du nouveau codeur implémenté sur la même carte FPGA (Xilinx-Virtex 5) sont présentés dans le tableau suivant :

	Fréquence de fonctionnement (MHz)	Temps de traitement (ms)	Puissance dynamique consommée (mW)	Puissance statique dissipée (mW)	Energie consommée (mJ)
Codeur étudié	113.84	17.84	270	5.79	4.81

Tableau 4.3 : Performances du codeur étudié : implémentation sur une carte FPGA Xilinx Virtex 5

Le tableau 4.3 montre que la puissance dynamique consommée est largement supérieure à la puissance statique lors de l'implémentation du codeur spécifié sur la carte FPGA Xilinx-Virtex 5. Le codeur consomme environ 4.8 mJ. Par contre, le nœud de réseau de capteurs sans fil (plateforme MICA2) consomme entre 436 mJ et 1376 mJ pour coder l'image. Le coût du traitement lié au codage varie en fonction du nombre de niveaux appliqués pour la transformée en ondelettes de Haar, le nombre d'itérations du codage EZW, et le seuil de quantification appliqué. L'implémentation matérielle sur carte FPGA a permis d'atteindre des gains énergétiques considérables pour le nœud source du réseau de capteurs sans fil (celui qui fait la compression). Le taux de compression élevé atteint par le codeur permettra également de bénéficier de gains énergétiques importants pour les autres nœuds du réseau (réduction du nombre de paquets échangés à travers le réseau).

Les performances du codeur peuvent être améliorées davantage dans le cas d'une éventuelle implémentation sur un circuit ASIC. Le tableau suivant présente les résultats de synthèse obtenus pour différentes technologies d'intégration :

	Technologie utilisée	F max (MHz)	Puissance dynamique (mW)	Surface du codeur (mm ²)
Codeur étudié	45 nm	790	20.76	0.065
	90 nm	392	23.17	0.17
	0.13 µm	347	31.87	0.22
	0.18 µm	215	44.13	0.59
	0.35 µm	139	57.60	2.31
	0.6 µm	78	75.74	11.27

Tableau 4.4 : Performances du codeur étudié : implémentations ASIC pour différentes technologies standards d'intégration

Les résultats présentés dans le tableau 4.4 montrent que l'implémentation sur ASIC du codeur étudié a permis d'atteindre des fréquences de traitement élevées, des puissances consommées assez faibles, et une petite surface occupée par le codeur, en comparaison avec les résultats obtenus lors de l'implémentation du même codeur sur une carte FPGA Xilinx-Virtex-5.

Le codeur étudié (*HWT – Quantification – EZW – Arithmétique modifié*) est légèrement moins performant que le premier codeur implémenté (*HWT – Quantification – EZW – RLE*) en termes de fréquence de fonctionnement, puissance dissipée et surface occupée. La fréquence maximale de fonctionnement du circuit est moins bonne ; la puissance dynamique consommée est légèrement supérieure et la surface du circuit excède un petit peu

celle du premier codeur. Les écarts de performances des deux codeurs sont très faibles, et sont dus principalement à la complexité du *codage arithmétique modifié* par rapport au codage *RLE* (chapitre 3 - section 3.2.2). Par contre, ce deuxième codeur est plus performant en termes de caractéristique "*débit-distorsion*" ainsi qu'aux gains énergétiques apportés sur la totalité du réseau de capteurs sans fil. En effet, le codeur étudié atteint des taux de compression plus élevés en comparaison avec le premier codeur, ce qui implique moins de paquets transmis à travers le réseau et par conséquent prolonger la durée de vie des nœuds du réseau.

Dans le but de valider l'approche de codage étudiée, les performances du présent codeur ont été comparées à celles de quelques implémentations matérielles du standard JPEG 2000. JPEG 2000 a été choisi puisqu'il fait partie de la même famille que notre codeur (les deux codeurs utilisent les transformées en ondelettes). Le tableau suivant détaille les résultats obtenus pour trois implémentations matérielles différentes du codeur JPEG 2000 en technologie CMOS standard 0.18 μm :

	Modrzyk [100]	Sanyo [101]	Chang [102]	Liu [103]
Surface du codeur (mm^2)	19	13	20.1	<20
Fréquence (MHz)	100	54	42	100
Puissance dynamique (mW)	2000	900	384	450

Tableau 4.5 : Résultats d'implémentations du codeur JPEG 2000 en technologie CMOS standard 0.18 μm

Les implémentations matérielles du codeur JPEG 2000 montrent que notre solution est plus efficace en termes de fréquence maximale de fonctionnement, surface occupée par le codeur, et puissance dynamique consommée. La comparaison est basée sur les résultats présentés dans les tableaux 4.4 (technologie CMOS standard 0.18 μm) et 4.5. La surface occupée par la solution de codage proposée est nettement inférieure à toutes les implémentations citées dans le tableau 4.5 (le rapport des surfaces varie entre 22 et 33). Notre codeur est 2 fois plus rapide que les codeurs étudiés dans [100, 103], 4 fois plus rapide que le codeur étudié dans [101], et 5 fois plus rapide que l'architecture présentée dans [102]. La puissance dynamique consommée par le circuit de compression que nous avons conçu est largement inférieure à celles consommées par toutes les solutions du tableau 4.5 (le rapport des consommations varie entre 8 et 45).

Malgré que JPEG 2000 atteigne des taux de compression très élevés tout en gardant une bonne qualité d'image à la reconstruction, la complexité algorithmique de ce codeur est loin de satisfaire aux besoins spécifiques des réseaux de capteurs sans fil. Ce type de réseaux nécessite une solution qui consomme moins d'énergie et occupe moins de surface (et par

conséquent coûte moins cher). Le codeur que nous avons proposé satisfait mieux les compromis évoqués dans cette étude (qualité de l'image reconstituée, surface du circuit, complexité algorithmique, vitesse de calcul, énergie dissipée, etc.).

4.1.4. Codeur pour la compression vidéo

Ce codeur (chapitre 3 - section 3.3) utilise la même chaîne de compression étudiée précédemment (première chaîne de compression). L'unique différence consiste à détecter et coder uniquement les sous-blocs de l'image courante qui ont changé de façon significative par rapport aux mêmes sous-blocs de la trame précédente. L'implémentation *"software"* de la technique proposée sur un nœud de réseau de capteurs sans fil (plateforme MICA2) a montré la supériorité de cette technique par rapport aux codeurs vidéo standards. La qualité des vidéos reconstruites est bonne et les gains énergétiques du nœud sont considérables. L'énergie dissipée et le temps de calcul liés au traitement de la vidéo ne sont pas négligeables. L'implémentation matérielle du codeur permettra d'optimiser ces deux paramètres, et par conséquent de minimiser la consommation totale du nœud et le délai de transfert des vidéos codées.

Le schéma bloc du codeur implémenté est illustré par la figure suivante :

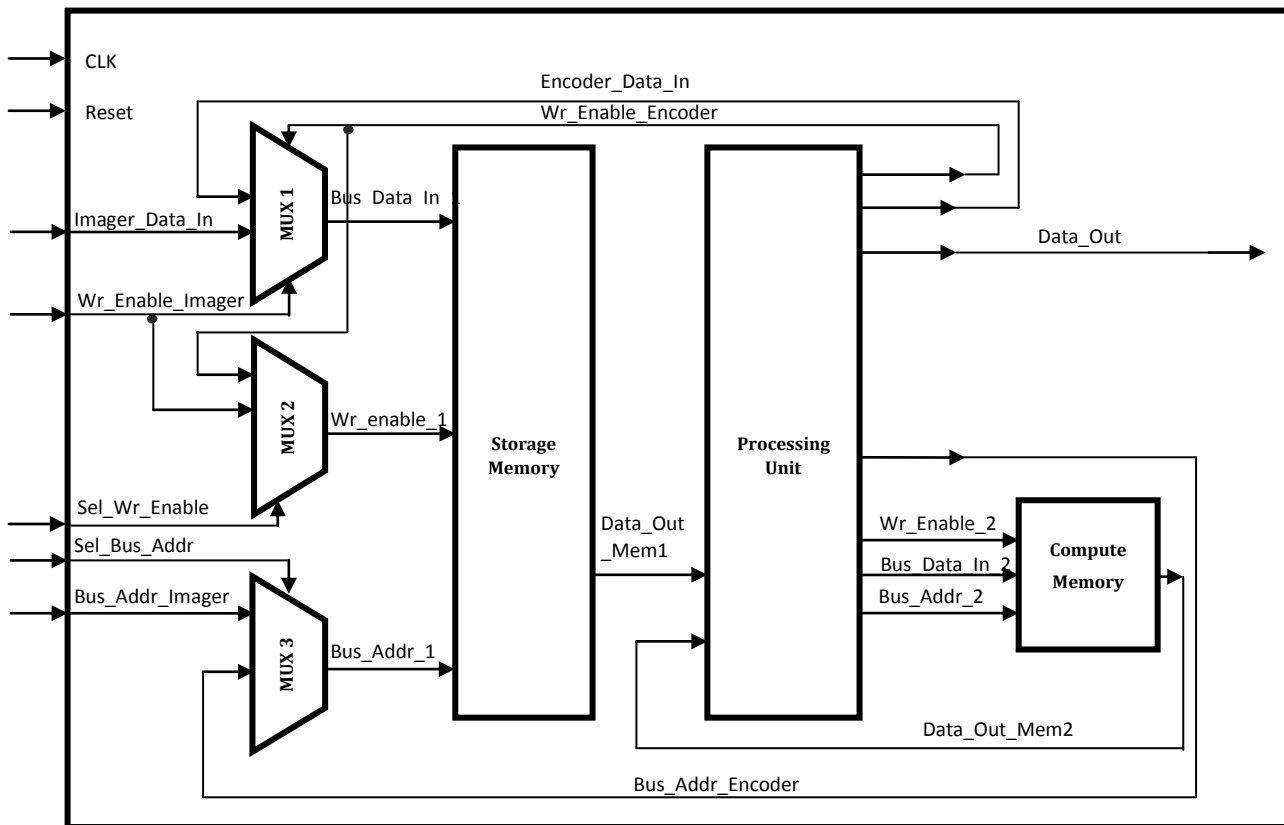


Figure 4.7 : Schéma bloc du codeur vidéo proposé

Le codeur intègre une unité de traitement, deux mémoires et trois multiplexeurs. La mémoire de stockage (mémoire à gauche) est accessible à la fois par l'unité de traitement et le convertisseur Analogique/Numérique en aval de l'imageur. Les trois multiplexeurs permettent de multiplexer les signaux en provenance de l'imageur et ceux provenant de l'unité de calcul. Au départ, le codeur stocke dans cette mémoire deux trames successives provenant du capteur d'image. L'unité de calcul traite les deux images stockées afin de détecter les blocs qui ont changé de façon significative d'une trame à l'autre. Ces blocs seront compressés en utilisant la chaîne de compression basée sur la *transformée en ondelettes de Haar*, le codage *EZW* et le codage *RLE*. La mémoire de traitement (mémoire à droite) est utilisée pour stocker les résultats temporaires liés au traitement de chacun des blocs à compresser. Cette mémoire n'est accessible que par l'unité de traitement. Elle est utilisée pour accélérer le traitement : la compression des blocs désignés par l'unité de traitement et l'acquisition d'une nouvelle image se font de façon simultanée. La compression de chacun des blocs désignés par l'unité de calcul est suivie par une étape de paquets du flux résultant. Les paquets seront par la suite transmis par le codeur à travers le réseau de capteurs sans fil. Le premier paquet envoyé lors du traitement de chaque bloc porte les identifiants de l'image (numéro de l'image dans la séquence vidéo) et du bloc traités (numéro du bloc dans l'image traitée). Ces informations sont indispensables à la reconstitution de la séquence vidéo par la suite.

Le tableau suivant présente les résultats de synthèse obtenus pour différentes technologies standards d'intégration lors de l'implémentation matérielle du codeur vidéo :

	Technologie utilisée	F max (MHz)	Puissance dynamique (mW)	Surface du codeur (mm ²)
Codeur étudié	45 nm	961	29.89	0.23
	90 nm	568	34.42	0.36
	0.13 µm	434	53.72	0.48
	0.18 µm	348	61.49	0.73
	0.35 µm	224	75.92	4.56
	0.6 µm	137	86.64	11.13

Tableau 4.6 : Performances du codeur vidéo : implémentations ASIC pour différentes technologies standards d'intégration

La fréquence maximale de fonctionnement du codeur implémenté varie entre 137 MHz et 961 MHz, en fonction de la technologie d'intégration utilisée. La puissance dynamique consommée varie entre 29 mW et 86 mW, et la surface du codeur varie entre 0.23 mm² et 11.13 mm². Ce codeur, contrairement aux deux autres codeurs étudiés précédemment,

permet d'exploiter à la fois la redondance spatiale et la redondance temporelle afin de coder efficacement les séquences vidéo. Le codeur fournit des résultats intéressants pour des séquences vidéo simples (caméra dont le champ de vision est fixe ou qui bouge peu, objets qui se déplacent lentement) telles que les vidéos utilisées en vidéo-surveillance. Le codeur est moins performant en termes de taux de compression et qualité d'image pour les vidéos rapides. Ceci n'est pas du tout gênant dans le cas particulier des réseaux de capteurs sans fil, qui ne sont pas conçus pour traiter et transmettre des vidéos rapides ou des vidéos hautes définitions.

Dans le but de mettre en évidence la présente technique de compression vidéo, l'approche devrait être comparée à des codeurs vidéo standards. Le tableau suivant présente des implémentations matérielles de quelques standards vidéo. Les implémentations ont été faites en utilisant la technologie CMOS standard 0.35 μm .

	F_{max} (MHz)	Surface du codeur (mm^2)	Puissance dissipée (mW)
MPEG2 [104]	54	6.79	400
MPEG2 [105]	54	155	1500
MPEG 4 (Shape Coding) [106]	23.5	NA	NA
MPEG 4 [107]	27	94.09	290
Le codeur proposé	224.72	4.56	75.92

Tableau 4.7 : Performances de quelques standards de compression vidéo implémentés en technologie CMOS standard 0.35 μm

L'approche de compression vidéo proposée est plus adaptée avec les nœuds de réseaux de capteurs sans fil. En fait, le codeur proposé consomme moins d'énergie et de surface que tous les codeurs cités dans le tableau 4.7. En plus, la fréquence de fonctionnement de notre circuit de compression dépasse largement celle des autres codeurs présentés. Cette comparaison prouve la supériorité de notre approche. Malgré cette supériorité, les performances du codeur étudié (taux de compression et qualité de la vidéo reconstruite) sont dégradées lorsqu'il traite des vidéos rapides. Dans ce cas-là, les codeurs vidéo standards (MPEG-x et H.26x) seront privilégiés.

4.2. Vers l'implémentation d'un réseau de capteurs de vision sans fil à ultra-faible consommation d'énergie

Certes, la conception d'un système de vision faible consommation et intégrant des fonctionnalités de compression, s'est avérée d'une grande utilité dans la conservation de

l'énergie des nœuds du réseau, mais nous sommes toujours loin du réseau de capteurs de vision sans fil idéal. En effet, l'optimisation de l'énergie consommée par les nœuds devrait être faite à plusieurs niveaux (consommation de chacun des composants du nœud, protocole de routage, technologie d'intégration, système d'exploitation, etc.). L'influence de chacun de ces niveaux sera détaillée dans cette section.

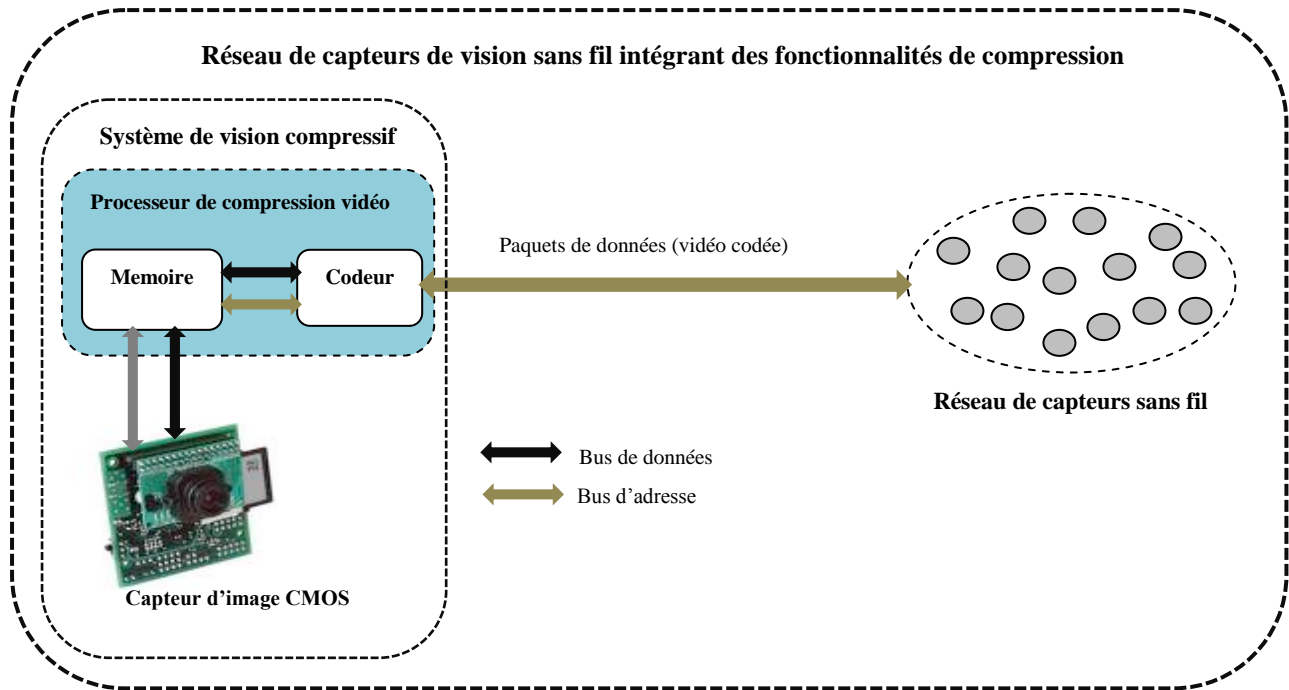


Figure 4.8 : Modèle d'un réseau de capteurs de vision sans fil

Dans ce travail de recherche, nous avons prouvé que la compression d'image permet de prolonger la durée de vie du réseau de capteurs de vision en réduisant le nombre de paquets transmis à travers ce réseau. Une attention particulière devrait être prise en compte lors du choix des techniques de compression et de codage à employer afin de mieux satisfaire le compromis entre la complexité algorithmique et le taux de compression atteint par le codeur. Malheureusement, les travaux menés ne représentent qu'une partie des tâches qui devraient être accomplies afin de réduire de façon optimale la consommation du système entier, représenté par la figure 4.8. En effet, la conception d'un système de vision compressif permet de réduire uniquement l'énergie liée aux transceivers des nœuds du réseau ; d'autres sources de consommation excessive de l'énergie nécessitent d'être étudiées afin de minimiser la consommation globale du réseau.

4.2.1. Optimisations liées au matériel

Malheureusement, les nœuds de réseaux de capteurs sans fil existants sur le marché incluent certains blocs qui consomment trop d'énergie. Ces blocs varient d'une plateforme à

une autre. Pour certaines plateformes, les calculs (et par conséquent le microcontrôleur) sont très coûteux en énergie, pour d'autres, c'est l'émetteur/récepteur radio qui affecte le plus la consommation des nœuds. Les lectures/écritures mémoire pourraient être aussi une source importante de consommation pour certaines plateformes [108].

Radio	Débit binaire (kbps)	Consommations typiques		
		Mode veille (μ A)	Mode transmission (mA)	Mode réception (mA)
CC1000	76.8	0.2	10.4 (0 dBm)	7.4/9.6
CC2420	250	1	17.4 (0 dBm)	18.8
nRF2401A	1000	0.9	13 (0 dBm)	19
nRF24L0A	2000	0.9	11.3 (0 dBm)	12.3

Tableau 4.8 : Consommations typiques de quelques transceivers de réseaux de capteurs sans fil

Microcontrôleur	Consommations typiques	
	Mode veille	Fonctionnement normal
Intel StrongARM	100 mW (veille)	400 mW
	50 μ W (veille prolongée)	
Texax Instruments MSP 430	6 μ W (veille)	1.2 mW
	0.3 μ W (veille prolongée)	
Atmem ATmega 128L	6 mW (veille)	15 mW
	75 μ W (veille prolongée)	

Tableau 4.9 : Consommations typiques de quelques microcontrôleurs utilisés au sein des nœuds de réseaux de capteurs sans fil

Mémoire	Consommations typiques	
	Mode lecture	Mode écriture
AT45DB041B (plateforme MICA)	9.4 mA	21.6 mA
ST M25P80 (plateforme Telos)	4.1 mA	15.1 mA

Tableau 4.10 : Consommations typiques lors de l'accès mémoire pour les plateformes de réseaux de capteurs sans fil, MICA et Telos

Les trois tableaux présentés ci-dessus (tableaux 4.8, 4.9 et 4.10) montrent bien la grande variation des consommations d'une plateforme à l'autre. Chaque plateforme de réseaux de capteurs sans fil intègre des blocs qui consomment peu d'énergie, mais inclut aussi d'autres blocs gourmands en énergie.

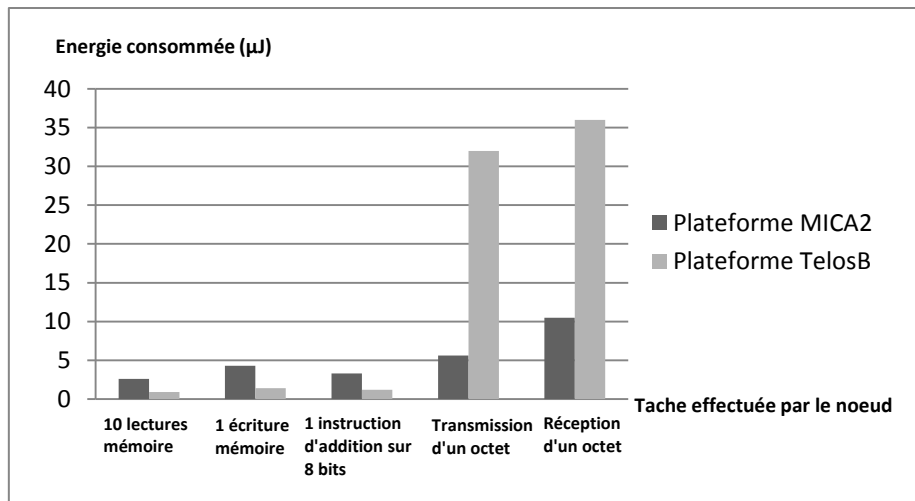


Figure 4.9 : Consommations typiques des nœuds de deux plateformes de réseaux de capteurs sans fil

La figure 4.9 présente quelques résultats expérimentaux extraits en utilisant les deux plateformes de réseaux de capteurs sans fil, MICA2 et TelosB. Ces résultats montrent que MICA2 est favorisée par rapport à TelosB en termes de consommation énergétique lorsqu'il s'agissait de transmettre ou recevoir des paquets de données. En effet, MICA2 utilise l'émetteur/récepteur radio CC1000 qui consomme moins d'énergie que le transcepteur CC2420 utilisé par TelosB. Par contre, les nœuds de type TelosB sont moins gourmands en énergie en mode "*traitement de données*". Cet avantage s'explique par le fait que TelosB est équipé d'un microcontrôleur *MSP430* et d'une mémoire *AT45DB041B*, deux composants qui ont des consommations réduites en comparaison avec le microcontrôleur *Atmega128L* et la mémoire *ST M25P80* de la plateforme Mica2.

L'idéal consiste à concevoir un nœud de réseau de capteurs sans fil dédié aux applications multimédias. Le nœud pourra intégrer certains blocs commercialisés existants sur le marché, mais aussi d'autres blocs qu'on pourra concevoir nous-même. L'ultra-faible consommation est un besoin fondamental pour les réseaux de capteurs de vision sans fil ; les différents blocs du nœud à concevoir devraient être optimisés afin de prolonger au maximum la durée de vie du réseau. L'utilisation de circuits asynchrones pourrait être très intéressantes dans le futur proche afin de prolonger au maximum la durée de vie des réseaux de capteurs de vision. Les technologies d'intégration les plus récentes pourraient être exploitées, mais une attention particulière devrait être prise en compte afin d'avoir un coût final et une puissance statique raisonnables.

4.2.2. Optimisations au niveau routage

Les protocoles utilisés actuellement par les réseaux de capteurs sans fil ne sont pas adaptés au cas particulier des applications liées au traitement et à la transmission d'images. Ce type d'applications nécessite un calcul intensif et un nombre important de paquets à échanger à travers le réseau. Les protocoles dédiés aux réseaux *Ad-hoc* nécessitent aussi un échange de paquets entre les nœuds voisins (mise à jour des tables de routage, requêtes et acquittements entre les nœuds et leurs *Cluster-Heads*, etc.). Les nœuds analysent puis traitent ces paquets afin de prendre des décisions ou envoyer des réponses à un ou plusieurs destinataires.

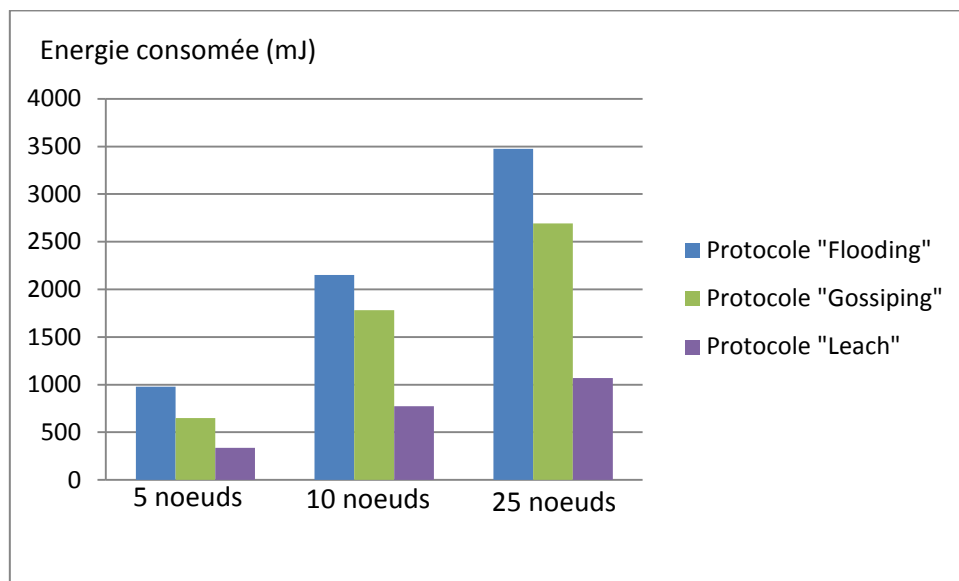


Figure 4.10 : Consommations énergétiques dues aux protocoles de routage lors de la transmission d'une image non compressée (image de taille 128x128 pixels codés sur 8 bpp)

La figure 4.10 montre la consommation moyenne d'un nœud de réseau de capteurs sans fil liée à chacun des protocoles de routage, "*Leach*", "*Gossiping*" et "*Flooding*" lors de la transmission d'une image de taille 128x128 pixels – 8 bpp. Le coût de transmission de l'image n'est pas pris en compte dans cette figure. L'objectif est de comparer les coûts énergétiques des protocoles utilisés. Nous avons justifié précédemment (chapitre 3 – section 3.1) la supériorité du protocole *Leach* par rapport aux deux autres protocoles étudiés en ce qui concerne la consommation énergétique totale (transmission des paquets liés au routage et ceux liés à l'image à transmettre) d'un nœud du réseau. Malgré cette supériorité, la consommation, directement liée au nombre de nœuds du réseau, reste relativement élevée. Pour un réseau de 25 nœuds, chaque nœud consomme plus de 1 joule pour communiquer avec ses voisins pendant le transfert d'une image vers le nœud puits. Dans [7], l'auteur a montré

que la consommation d'un nœud *MICA2* attaché à une caméra *Cyclops*, lors de la transmission directe (sans protocole de routage) d'une image de taille 128x128 pixels codés sur 8 bpp, est de 181 mJ. L'acquisition de l'image (caméra *Cyclops*) coûte 90 mJ alors que la transmission de l'image par le nœud *MICA2* consomme 91 mJ. Malheureusement, la réalité est tout autre ; un réseau dense nécessite l'emploi d'un protocole de routage qui nécessite lui aussi de l'énergie pour se mettre en place. La transmission d'une image vers un nœud puits nécessite beaucoup plus d'énergie à cause des retransmissions de paquets, dues aux collisions et aux pertes de données. En plus, les échanges de données liés au protocole de routage ont un coût considérable. Pour un réseau de capteurs sans fil de 25 nœuds employant le protocole de routage *Leach*, chaque nœud du réseau consomme entre 45 et 48 fois plus d'énergie que dans le cas particulier du transfert direct d'une image (sans protocole de routage et sans retransmissions). Cet écart important soulève la question de l'utilisation des protocoles de routage usuels dédiés aux réseaux de capteurs sans fil. Il pourrait donc s'avérer nécessaire de recourir à la conception d'un protocole de routage adapté aux applications multimédias [109]. Cette tâche pourrait être validée soit en apportant des modifications sur l'un des protocoles existants, soit par la conception d'un nouveau protocole qui répond mieux aux spécificités des applications liées au traitement et au transfert d'images.

4.3. Conclusion

Dans ce chapitre, nous avons montré l'intérêt de l'implémentation matérielle de chacune des techniques de compression adoptées, par rapport aux implémentations logicielles étudiées dans le chapitre précédent. Les gains énergétiques sont plus importants, et permettent de prolonger davantage la durée de vie des nœuds sources (ceux qui effectuent la compression des images). Chaque nœud source devrait, donc, être équipé d'un capteur d'image CMOS intégrant une circuiterie capable de réduire au maximum le flux de données à transmettre à travers le réseau. Certes, les solutions proposées sont bénéfiques en termes de conservation de l'énergie dissipée à la fois par les nœuds sources et par la totalité du réseau, mais ces circuits ont un coût supplémentaire considérable. Il existe donc un compromis entre l'optimisation de la consommation énergétique des nœuds du réseau et le coût des solutions envisagées.

Conclusion Générale et Perspectives

Ce manuscrit de thèse synthétise les travaux menés par notre équipe de recherche ainsi que la communauté scientifique internationale ces dernières années dans les domaines de réseaux de capteurs de vision sans fil et des systèmes de vision intelligents intégrant des traitements sur le plan focal de l'imageur. Malheureusement, les deux axes de recherche ont été étudiés séparément par les scientifiques : certaines équipes de recherche ont présenté des contributions logicielles (algorithmes implémentés au sein des nœuds du réseau) à efficacité réduite pour faire face aux contraintes spécifiques des réseaux de capteurs de vision sans fil ; d'autres équipes de recherche ont essayé de concevoir des systèmes de vision intelligents et à faible coût, permettant de minimiser les flux de données en sortie de l'imageur. Le premier groupe de chercheurs a opté pour des plateformes existantes de capteurs d'image, généralement trop gourmandes en énergie et non optimisées pour des applications multimédias dédiées aux réseaux de capteurs sans fil. Leurs travaux consistent à acquérir des images, les stocker, puis les compresser localement ou de façon distribuée à travers le réseau. Les nœuds du réseau ont généralement des ressources limitées (capacités de calcul, mémoire, réserves énergétiques, etc.) qui ne permettent pas d'implémenter des algorithmes assez complexes, et par conséquent très efficaces en termes de gain énergétique. Le deuxième groupe de chercheurs nous présente des contributions liées aux capteurs d'image intégrant des fonctionnalités de compression et de détection de mouvement sur le plan focal de l'imageur. Malheureusement, ces travaux sont dé-corrélés des travaux liés aux réseaux de capteurs de vision sans fil ; aucune considération des contraintes strictes de ce type de réseaux n'est prise en compte lors de la conception de ces systèmes de vision (coût supplémentaire engendrée par le capteur d'image, puissance dissipée, surface occupée par l'imageur, efficacité des techniques de compression utilisées en relation avec les nœuds du réseau, etc.).

Dans ce travail de recherche, nous avons regroupé les deux approches (travaux liés aux capteurs de vision compressifs et travaux liés aux réseaux de capteurs sans fil intégrant des fonctionnalités de compression à la source) afin de trouver une solution adéquate qui permettrait de prolonger au maximum la durée de vie du réseau. Tout d'abord, nous avons commencé par l'étude du transfert d'images entières, sans compression, à travers le réseau. Les consommations énergétiques des nœuds du réseau sont trop élevées ; il paraît donc évident de procéder à l'implémentation de techniques de compression efficaces pour réduire les flux de données échangés à travers le réseau. Une attention particulière devrait être prise

en compte lors du choix des codeurs à implémenter : l'énergie dissipée pour coder les images ne doit pas excéder l'énergie conservée en transmettant l'image ou la vidéo compressée. Les codeurs sélectionnés ont permis, grâce à leurs complexités réduites ainsi que leurs taux de compression élevés, d'atteindre des gains énergétiques considérables à la fois pour les nœuds sources ainsi que pour la totalité du réseau.

Les codeurs, implémentés de façon logicielle sur un ou plusieurs nœuds du réseau, ont des consommations énergétiques considérables. Les nœuds sources consomment trop d'énergie pour coder les images avant de les transmettre à travers le réseau. Ces nœuds épuisent rapidement leurs batteries puisqu'ils exécutent des algorithmes de compression relativement complexes. L'implémentation matérielle de ces techniques de compression et de codage a permis d'optimiser l'énergie liée au traitement. Les résultats, présentés et discutés dans le dernier chapitre ont permis de valider cette approche. Dans ce chapitre, nous avons aussi détaillé l'architecture de l'imageur CMOS choisi, susceptible de remplacer les caméras commercialisées utilisées par la communauté scientifique, et qui sont généralement non optimisées pour des applications multimédias à travers les réseaux de capteurs sans fil. L'ensemble formé par le capteur d'image et le codeur vidéo présente un système de vision intelligent, et à faible consommation énergétique, capable de réduire le flux de données échangé à travers le réseau au prix d'un coût raisonnable (énergie dissipée et surface silicium). Ce système de vision pourrait être intégré par la suite au sein des nœuds du réseau afin d'avoir un seul dispositif compact.

En ce qui concerne ce travail de recherche, il reste à étudier plusieurs aspects importants liés aux réseaux de capteurs de vision sans fil :

- Concevoir un convertisseur Analogique/Numérique (pour le capteur d'image) à faible consommation énergétique. En effet, la conversion A/N est très coûteuse en énergie pour les capteurs d'image CMOS. L'optimisation efficace de l'énergie consommée par l'imageur nécessite la conception d'un CAN à ultra-faible consommation énergétique.
- La conception d'un nœud de réseau de capteurs sans fil fortement optimisé en termes d'énergie consommée. En effet, les nœuds de réseaux de capteurs sans fil existants sur le marché incluent toujours un ou plusieurs blocs qui consomment trop d'énergie. Ces blocs varient d'une plateforme à une autre. Pour certaines plateformes, les traitements et les calculs sont très coûteux en énergie, pour d'autres, c'est le transceiver radio qui consomme le plus d'énergie. Les lectures/écritures mémoire pourraient être aussi une

source importante de consommation pour certaines plateformes. L'optimisation de l'énergie consommée par les nœuds du réseau passe par la réduction des puissances dissipées au sein de chaque bloc fonctionnel des nœuds du réseau (transceiver, microcontrôleur, accès mémoire, capture de l'image, etc.).

- Etudier l'influence des protocoles utilisés par ce type de réseaux sur sa durée de vie. La conception de nouveaux protocoles (routage, accès au canal, etc.) adaptés aux applications multimédias à travers les réseaux de capteurs sans fil pourrait être envisagée en cas de nécessité (si les protocoles actuels occupent trop de ressources : énergie, mémoire, temps).

Ce manuscrit de thèse ne présente qu'une modeste contribution ayant pour objectif de faire face aux contraintes strictes des réseaux de capteurs de vision sans fil. Les futurs travaux de notre équipe de recherche permettraient d'aller plus loin, dans l'objectif d'avoir un réseau de capteurs d'image ayant à la fois une qualité de service notable et une très longue durée de vie.

LISTE DES REFERENCES

- [1] Aurélien Buhrig, Optimisation de la consommation des nœuds de réseaux de capteurs sans fil, thèse de doctorat, Institut National Polytechnique de Grenoble, Avril 2008.
- [2] Changsu Suh, Zeeshan Hameed Mir, Young-Bae Ko. Design and implementation of enhanced IEEE 802.15.4 for supporting multimedia service in Wireless Sensor Networks. Elsevier, Computer Networks, 2008; 52:2568-2581.
- [3] Antoine Gallais, Ordonnancement d'activité dans les réseaux de capteurs: l'exemple de la couverture de surface, thèse de doctorat, Université des sciences et technologies de Lille, Juin 2007.
- [4] Ludovic Samper, Modélisations et analyses de réseaux de capteurs, thèse de doctorat, Institut National Polytechnique de Grenoble, Avril 2008.
- [5] Amine Abbas, Les réseaux sensoriels sans fil : Etat de l'art et perspectives de recherche, équipe de recherche AND du LIFC, Belfort, laboratoire d'informatique de l'université de Franche-comté, DNA Research day, May 2005.
- [6] François Ingelrest, Protocoles localisés de diffusion et économie d'énergie dans les réseaux ad hoc et de capteurs, thèse de doctorat, Université des sciences et technologies de Lille, Juin 2006.
- [7] Cristian Duran-Faundez, Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie, thèse de doctorat, Université Henri Poincaré, Centre de Recherche en Automatique de Nancy, Juin 2009.
- [8] Holger Karl, Andreas Willing. Protocols and Architectures for Wireless Sensor Networks. Book published by Wiley & Sons, June 2005.
- [9] Clément Saad, Quelques contributions dans les réseaux de capteurs sans fil, thèse de doctorat, Université d'Avignon et des Pays de Vaucluse, Juillet 2008.
- [10] Rahim Kacimi, Techniques de conservation d'énergie pour les réseaux de capteurs sans fil, thèse de doctorat, Institut National Polytechnique de Toulouse, Septembre 2009.
- [11] Hela Boussetta, Modélisation multi-physiques et simulations globales de systèmes autonomes sur puce, thèse de doctorat, Université de Grenoble et Faculté des sciences de Bizerte, Février 2010.
- [12] Fabrice Valois, Auto-organisation de réseaux radio multi-sauts, habilitation à diriger les recherches, INSA de Lyon et l'Université Claude Bernard - Lyon 1, Novembre 2007.
- [13] Julien Vaudour, Vincent Gauthier, Comparaison de différentes couches MAC pour les réseaux de capteurs, Rapport de recherche INT N° 06006RST GET/INT UMR 5157 SAMOVAR, Institut National des Télécommunications, Evry, France, 2005.
- [14] Mickael Cartron, Vers une plate-forme efficace en énergie pour les réseaux de capteurs sans fil, thèse de doctorat, Université de Rennes 1, Décembre 2006.
- [15] Kamal Beydoun, Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs, thèse de doctorat, L'UFR des Sciences et Techniques de l'Université de Franche-Comté, Décembre 2009.
- [16] Alexandre Delye de Mazieux, Vincent Gauthier, Michel Marot, Monique Becker, Etat de l'art sur les réseaux de capteurs, Rapport de recherche INT N° 05001RST GET/INT UMR 5157 SAMOVAR, Institut National des Télécommunications, Evry, France, 2005.
- [17] Hung-Cuong LE, Optimisation d'accès au médium et stockage de données distribuées dans les réseaux de capteurs, thèse de doctorat, L'UFR des sciences et technique de l'université de Franche-comté, Octobre 2008.
- [18] Yacine Challal, Réseaux de capteurs sans fil, Cours: systèmes intelligents pour le transport, Université de technologie de Compiègne, Novembre 2008.
- [19] Arthur Spivak, Adam Temam, Alexander Belenky, Orly Yadid-Pecht, Alexander Fish (2012), Low-Voltage 96 dB Snapshot CMOS image sensor with 4.5 nW power dissipation per pixel, Sensors, 12(8), 10067-10085.

- [20] Manuel Camus, Architecture de réception RF très faible coût et très faible puissance. Application aux réseaux de capteurs et au standard ZigBee, thèse de doctorat, Université de Toulouse, Février 2008.
- [21] Ala Qadi, Steve Goddard, Shane Farritor, A dynamic voltage scaling algorithm for sporadic tasks, In proceedings of the 24th IEEE real-time systems symposium, Cancun, Mexico, December 2003, pp. 52-62.
- [22] Gil de Sousa, Etude en vue de la réalisation de logiciels bas niveau dédiés aux réseaux de capteurs sans fil: microsystème de fichiers, thèse de doctorat, Université Blaise Pascal - Clermont 2, Octobre 2008.
- [23] Philip Levis, TinyOS Programming, Book published by Cambridge University Press, June 2006.
- [24] Philip Levis, Nelson Lee, TOSSIM: a simulator for TinyOS network, UC Berkeley, September 2003.
- [25] Philip Levis, Nelson Lee, Matt Welsh, David Culler. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. SenSys'03 Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, November 2003.
- [26] Huaming Wu, Alhussein A. Abouzeid. Energy efficient distributed image compression in resource-constrained multihop wireless networks. Elsevier, Computer Communications, 2005; 28:1658-1668.
- [27] Ching-Wen Chen, Chuan-Chi Weng and Chang-Jung Ku. Design of a low power and low latency MAC protocol with node grouping and transmission pipelining in wireless sensor networks. Vehicular Technology Conference, 2008.VTC Spring 2008. IEEE, (2008) p. 143-147.
- [28] Jorge Fernandez, Berni Ricardo Carmona, Galan Angel Rodriguez, Vazquez, Vision-enabled WSN nodes: state of the art, Low-power smart imagers for vision-enabled sensor networks: chapter 2, Springer New York Heidelberg Dordrecht London, DOI 10.1007/978-1-4614-2392-8, 2012.
- [29] Cristian Duran-Faundez, Vincent Lecuire, Francis Lepage, Tiny block-size coding for energy-efficient image compression and communication in wireless camera sensor networks. In Signal Processing Image Communication, 2011; 26:466-481.
- [30] Tossaporn Srisooksai, Kamol Keamarungsi, Poonlap Lamsrichan, Kiyomichi Araki. Practical data compression in wireless sensor networks: A survey, Journal of Network and Computer Applications, 2012; 35:37-59.
- [31] Jun Ohta, Smart CMOS Image Sensors and Applications, Volume 129 of Optical Science and Engineering, CRC Press by Taylor and Francis Group, 2008, ISBN 13 :978-0-8493-3681-2, 267 pages.
- [32] Moncef Mekouar, Compression d'images médicales par ondelettes et régions d'intérêt, Maitrise en technologie des systèmes, Ecole de technologie supérieure, Montréal, Juin 2001.
- [33] Huaming Wu, Alhussein A. Abouzeid, Error resilient image transport in wireless sensor networks, Elsevier, Computer Networks, 2006; 50:2873-2887.
- [34] Min Wu, Chang Wen Chen, Multiple bitstream image transmission over wireless sensor networks, In: Sensors, october 2003, Proceedings of IEEE (Vol. 2, pp. 727-731). IEEE.
- [35] Vincent Lecuire, Cristian Duran-Faundez, Nicolas Krommenacker. Energy-efficient transmission of wavelet-based images in wireless sensor networks. EURASIP J. Image Video Process 2007; Article ID. 47345, 11 pages.
- [36] K. S. THYAGARAJAN, Still Image and Video Compression with MATLAB, Published by JOHN WILEY & SONS, INC, 2011, ISBN 978-0-470-88692-2, 442 pages.
- [37] Kaddachi Lassaad, Soudani Adel, Lecuire Vincent, Tourki Kholdoun, Mekkaoui Leila, Moureaux. Jean-Marie. Low power hardware-based image compression solution for wireless camera sensor networks. Computer Standards and Interfaces, 2012; 34:14-23.
- [38] Mekkaoui Leila, Vincent Lecuire, Jean-Marie Moureaux, Camera Sensor Networks, Published in the 2nd International Conference on Image Processing Theory, Tools and applications, IPTA 2010.
- [39] Huaming Wu, Alhussein A. Abouzeid, Power aware image transmission in energy constrained wireless networks, In Computers and Communications, 2004, Proceedings ISCC 2004, 9th international symposium on (Vol. 1 pp. 202-207). IEEE.
- [40] Huaming Wu, Alhussein A. Abouzeid, Energy efficient distributed JPEG 2000 image compression in multihop wireless networks, Proc. of IEEE Workshop on Applications and Services in Wireless Networks, 2004, pp. 152-160.

- [41] David Slepian, Jack K. Wolf (1973), Noiseless Coding of Correlated Information Sources, *Information theory*, IEEE Transactions on, 19(4), 471-480.
- [42] Raymond Wagner, Robert Nowak, Richard Baraniuk. Distributed image compression for sensor networks using correspondance analysis and super-resolution. in: *Proceeding of IEEE International Conference on Image Processing (ICIP)*, 2003; 1:597-600.
- [43] Qin Lu, Wusheng Luo, Jidong Wang, Bo Chen. Low complexity and energy efficient image compression scheme for wireless sensor networks. *Elsivier, Computer Networks*, 2008; 52 (13): 2594-2603.
- [44] Zhi-Yan Cao, Zheng Zhou Ji, Ming Zeng Hu. An image sensor node for wireless sensor networks. *Information Technology: Coding and Computing*, 2005; 2:740-745.
- [45] Eugenio Culurciello, Andreas G. Andreou, CMOS image sensors for sensor networks, *Analog Integr Circ Sig Process*, 49 (2006) 39-51.
- [46] Leonardo Gasparini, Roberto Manduchi, Massimo Gottardi, Dario Petri (2011), An Ultra-Low Power Wireless Camera Node: Development and Performance Analysis, *Instrumentation and Measurement*, IEEE Transactions on, 60 (12), 3824-3832.
- [47] Milin Zhang, Amine Bermak, CMOS Image Sensor with On-Chip Image Compression : A Review and Performance Analysis, *Hindawi Publishing Corporation, Journal of Sensors*, 2010, Article ID. 920693, 17 pages.
- [48] Akos Zarandy, *Focal-Plane Sensor-Processor Chips*, Springer Science + Business Media, 2011, ISBN 13 :978-1-4419-6475-5, 320 pages.
- [49] Hugo L. Haas, José Gabriel R. C. Gomes, Antonio Petraglia, Analog inner product operations for image compression in 0.35- μ m CMOS, *Analog Integr Circ Sig Process*, 57 (2008) 141-150.
- [50] Tristan Thabuis, *Imageurs numériques basse consommation exploitant les caractéristiques du flux vidéo*, thèse de doctorat, Institut National Polytechnique de Grenoble, Avril 2010.
- [51] Estelle Labonne, *Contributions à la conception de capteurs de vision CMOS à grande dynamique*, thèse de doctorat, Institut National Polytechnique de Grenoble, Juillet 2007.
- [52] Gérome Goy, *Etude, conception, et réalisation d'un capteur d'image APS en technologie standard CMOS pour des applications faible flux de type viseur d'étoiles*, thèse de doctorat, Institut National Polytechnique de Grenoble, 2002.
- [53] Fabrice Gensolen, *Architecture et conception de rétines CMOS: intégration de la mesure du mouvement global dans un imageur*, thèse de doctorat, Université Montpellier 2, Septembre 2006.
- [54] David Navarro, *Architecture et conception de rétines silicium CMOS: Application à la mesure du flot optique*, thèse de doctorat, Université Montpellier 2, Octobre 2003.
- [55] Gérôme Dubois, *Conception en technologie CMOS d'un système de vision dédié à l'imagerie rapide et aux traitements d'images*, thèse de doctorat, Université de Bourgogne, Août 2008.
- [56] Livier Lizarraga, *Technique d'auto test pour les imageurs CMOS*, thèse de doctorat, Institut National Polytechnique de Grenoble, Novembre 2008.
- [57] Zhimin Zhou, Bedabrata Pain, Eric R. Fossum, CMOS Active Pixel Sensor with On-Chip Successive Approximation Analog-To-Digital Converter, *IEEE Transactions on Electron Devices*, October 1997, 44(10) (1997) 1759-1763.
- [58] Karine Matou, *Capteur d'image logarithmique CMOS avec compensation "On-chip" du bruit spatial fixe*, thèse de doctorat, Université Paris XI, Juillet 2003.
- [59] Guillaume Demésy, *Modélisation électromagnétique tri-dimensionnelle de réseaux complexes. Application au filtrage spectral dans les imageurs CMOS*, thèse de doctorat, Université Paul Cézanne, Aix-Marseille III, Avril 2009.
- [60] Sunetra K. Mendis, Sabrina E. Kemeny, Russell C. Gee, Bedabrata Pain, Craig O. Staller, Quiesup Kim, Eric R. Fossum, CMOS Active Pixel Image Sensors for Highly Integrated Imaging Systems, *IEEE Journal of Solid-State Circuits*, February 1997, 32 (2) (1997) 187-197.
- [61] S. K. Mendis, S. E. Kemeny, R. C. Gee, B. Pain, C. O. Staller, Q. Kim, E. R. Fossum, CMOS active pixel image sensors for highly integrated imaging systems, *IEEE Journal of Solid-State Circuits*, Vol. 32, No. 2, pp. 187-197, Février 1997.

- [62] Delphine Emzivat, Claude Gagnadre, Eric Martin, Méthode de conception de l'architecture d'un capteur de vision dédié au contrôle qualité, 17ème Colloque GRETSI, Vannes, pp. 79-82, Septembre 1999.
- [63] Krzysztof Iniewski, Circuits at the nanoscale: Communications, Imaging and Sensing, part V: Circuits for Imaging and Sensing, CRC Press, ISBN-10: 1420070622, ISBN-13: 978-1420070620, September 2008.
- [64] Jungwon Lee, Efficient image compression system with a CMOS transform imager, In partial fulfillment of the requirements for the Degree Doctor of Philosophy in Electrical and Computer Engineering, School of Electrical and Computer Engineering, Georgia Institute of Technology, December 2009.
- [65] Kaddachi Lassaad, Soudani Adel, Lecuire Vincent, Mekkaoui Leila, Moureaux. Jean-Mari, Tourki Kholdoun, Design and performance analysis of a zonal DCT-based image encoder for wireless camera sensor networks, *Microelectronics Journal*, 43(2012), 809-817.
- [66] Shoushun Chen, Amine Bermak, Yan Wang, A CMOS Image Sensor with On-Chip Image Compression based on Predictive Boundary Adaptation and Memoryless QTD Algorithm, *IEEE Transactions on Very Large Scale Integration (VLSI)*, April 2011, 19 (4) (2011) 538-547.
- [67] Zhiqiang Lin, Michael W. Hoffman, Walter D. Leon, Nathan Schemm, Sina Balkir, A CMOS Image Sensor with Focal Plane SPIHT Image Compression, *IEEE International Symposium on Circuits and Systems (ISCAS 2008)*, May 2008, pp. 2134-2137.
- [68] Shoji Kawahito, Makoto Yoshida, Masaaki Sasaki, Kejiro Umehara, Daisuke Miyazaki, Yoshiaki Tadokoro, Kenji Murata, Shirou Doushou, Akira Matsuzawa, A CMOS Image Sensor with Analog Two-Dimensional DCT-Based Compression Circuits for One-Chip Cameras, *IEEE Journal of Solid-State Circuits*, December 1997, 32 (12) (1997) 2030-2041.
- [69] Edwin J. Tan, Zeljko Ignjatovic, Mark F. Bocko, A CMOS Image Sensor with Focal Plane Discrete Cosine Transform Computation, *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2007, pp. 2395-2398.
- [70] Abhishek Bandyopadhyay, Jungwon Lee, Ryan Robucci, Paul Hasler, A 80 μ W/frame 104x128 CMOS imager front end for JPEG compression, *IEEE International Symposium on Circuits and Systems (ISCAS 2005)*, May 2005, 5 (2005) 5318-5321.
- [71] Ashkan Olyaei, Roman Genov, Mixed-Signal CMOS Wavelet Compression Imager Architecture, 48th Midwest Symposium on Circuits and Systems, August 2005, 2 (2005) 1267-1270.
- [72] Canaan S. Hong, Richard Hornsey, On-Chip Binary Image Processing with CMOS Image Sensors, *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications III (Proceedings of SPIE Volume 4669)*, April 2002, pp. 125-136.
- [73] Evgeny Artyomov, Yair Rivenson, Guy Levi, Orly Yadid-Pecht (2005), Morton (Z) Scan Based Real-Time Variable Resolution CMOS Image Sensor. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(7), 947-952.
- [74] Sabrina E. Kemeny, Roger Panicacci, Bedabrata Pain, Larry Matthies, Eric R. Fossum, Multiresolution Image Sensor, *IEEE Transactions on Circuits and Systems for Video Technology*, August 1997, 7 (4) (1997) 575-583.
- [75] La rétine connexionniste: un nouveau système de vision sur puce (VSOC), *La revue trimestrielle du réseau Ecrin - n° 58*, Janvier 2005.
- [76] A. Elouardi, S. Bouaziz, A. Dupret, L. Lacassagne, J.O. Klein, R. Reynaud, Evaluation des performances d'un système de vision à base d'un capteur intelligent, *Institut d'électronique fondamentale, Université Paris Sud*.
- [77] Stuart Kleinfelder, SukHwan Lim, Xinqiao Liu, Abbas El Gamal, A 10 000 Frames/s CMOS Digital Pixel Sensor, *IEEE Journal of Solid-State Circuits*, December 2001, 36 (12) (2001) 2049-2059.
- [78] Abbas El Gamal, David Yang, Boyd Fowler, Pixel Level Processing – Why, What and How ?, In *Proceedings of the SPIE Electronic Imaging '99*, 1999, 12 pages.
- [79] Dominique Gin hac, Jérôme Dubois, Michel Paindavoine, Barthélémy Heyrman, A SIMD Programmable Vision Chip with High Speed Focal Plane Image Processing, *EURASIP Journal on Embedded Systems*, Special issue on design and architectures for signal and image processing, January 2008, Volume 2008, Article No. 8, 11 pages.

- [80] Zhiqiang Lin, Michael W. Hoffman, Nathan Schemm, Walter D. Leon, Sina Balkir, A CMOS Image Sensor for Multi-Level Focal Plane Image Decomposition, *IEEE Transactions on Circuits and Systems*, 55 (9) (2008) 2561-2572.
- [81] Qiang Luo, John G. Harris, A Novel Integration of On-Sensor Wavelet Compression for a CMOS Imager, *International Symposium on Circuits and Systems (ISCAS)*, 2002, pp. 325-328.
- [82] K. Aizawa, Y. Egi, T. Hamamoto, Y. Otsuka, M. Hatori, Focal Plane Compression Sensors Based on Pixel Parallel and Column Parallel Architectures, *IAPR Workshop on Machine Vision Applications (MVA'96)*, Tokyo, Japan, 1996, pp. 418-421.
- [83] Hawraa Amhaz, Hassan Abbass, Hakim Zimouche, Gilles Sicard, An Improved Smart Readout Technique Based on Temporal Redundancies Suppression Designed for Logarithmic CMOS Image Sensor, *18th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, December 2011, pp. 472-475.
- [84] Hawraa Amhaz, Gilles Sicard, X-axis Spatial Redundancy Supression : Contribution to the Integration of Smart Reading Techniques in a Standard CMOS Vision Sensor, *17th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, December 2010, pp. 311-314.
- [85] Hawraa Amhaz, traitement d'images bas niveau intégré dans un capteur de vision CMOS, thèse de doctorat, Institut National Polytechnique de Grenoble, Juillet 2013.
- [86] Alireza Nilchi, Joseph Aziz, Roman Genov, Focal Plane Algorithmically-Multiplying CMOS Computational Image Sensor, *IEEE Journal of Solid-State Circuits*, June 2009, 44 (6) (2009) 1829-1839.
- [87] Yu M. Chi, Ralph Etienne-Cummings, Gert Cauwenberghs, Image Sensor with Focal Plane Change Event Driven Video Compression, *IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2008, pp. 1862-1865.
- [88] Masatoshi Nishimura, Jan Van Der Spiegel, A CMOS Image Processing Sensor for the Detection of Image Features, *Analog Integrated Circuits and Signal Processing*, 45 (2005) 263-279.
- [89] Kenneth Barr, Krste Asanovic, Energy Aware Lossless Data Compression, *The First International Conference on Mobile Systems, Applications and Services*, San Francisco, CA, May 2003, pp. 231-244.
- [90] Ahmed Chefi, Adel Soudani, Gilles Sicard, Contribution to the design of a CMOS image sensor with low-complexity video compression for wireless sensor networks, *Journal of Systems Architecture*, <http://dx.doi.org/10.1016/j.sysarc.2013.07.010>.
- [91] Ahmed Chefi, Adel Soudani, Gilles Sicard, Hardware compression scheme based on low complexity arithmetic encoding for low power image transmission over WSNs, *International Journal of Electronics and Communications*, <http://dx.doi.org/10.1016/j.aeue.2013.08.006>.
- [92] Jason L. Hill, David E. Culler, MICA: a wireless platform for deeply embedded networks, *IEEE Computer Society*, 22 (6), pp. 12-24, November 2002.
- [93] Wendi Rabiner Heinzelman, Anantha Chandrakasan, Hari Balakrishnan, Energy efficient communication protocol for wireless microsensor networks, *Proceedings of the 33rd Hawai International Conference on System Sciences*, 2000.
- [94] L. Ferrigno, S. Marrano, V. Paciello, A. Pietrosanto, Balancing computational and transmission power consumption in wireless image sensor networks, *VECIMS 2005, IEEE international conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, Giardini Naxos, Italy, July 2005.
- [95] B. Heyne, C. C. Sun, J. Goetze, S. J. Ruan (2006, September), A computationnaly efficient high-quality cordic based DCT, In *European Signal Processing Conference (EUSIPCO 2006)*, Proceedings, Florence, Italy.
- [96] Mekkaoui Leila, Vincent Lecuire, Moureaux. Jean-Mari, Efficacité énergétique d'une DCT zonale rapide dans le contexte de la compression d'image dans les réseaux de capteurs sans fil (2010), *Compression et représentation des signaux audiovisuels, CORESA 2010*, 117-121.
- [97] Emmanuelle Bournay Bouchereau, Analyse d'images par transformées en ondelettes. Application aux images sismiques, thèse de doctorat, Université Joseph Fourier - Grenoble 1, Mars 1997.
- [98] Patrick Bas, Compression d'images fixes et de séquences vidéo, Cours ENSERG/INPG, Institut National Polytechnique de Grenoble, 2002.

- [99] Junaid Jameel Ahmad, Hassan Aqeel Khan, Syed Ali Khayam, Energy Efficient Video Compression for Wireless Sensor Networks, 43rd Annual Conference on Information Sciences and Systems (CISS), March 2009, pp. 629-634.
- [100] Damian Modrzyk, Michal Staworko. A high performance architecture of JPEG 2000 encoder. 19th European Signal Processing Conference (EUSIPCO 2011), Barcelona, Spain, August 29 - September 2, 2011.
- [101] Yamauchi Hideki, Okada Shigeyuki, Taketa Kazuhiko, Matsuda Yuh, Mori Tsugio, Watanabe Tsuyoshi, Matsuo Yoshihiro, Matsushita Yoshifumi, Sanyo Electr. Co. Ltd. 1440 x 1080 pixel, 30 frames per second motion-JPEG 2000 Codec for HD-movie transmission. ISSCC 2004 Solid-State Circuits Conference, San Francisco, 2004; 1:326-330.
- [102] Yu-Wei Chang, Chih-Chi Cheng, Chun-Chia Chen, Hung-Chi Fang, Liang-Gee Chen. 124 MSamples/s Pixel Pipelined Motion-JPEG 2000 Codec Without Tile Memory. IEEE Trans. on Circ. and S. for Video Tech, 2007; 17 (4):398-406.
- [103] Leibo Liu, Ning Chen, Hongying Meng, Li Zhang, Zhihua Wang, Hongyi Chen. A VLSI Architecture of JPEG 2000 Encoder. IEEE J. of Solid-State Circuits, 2004; 39 (11):2032-2040.
- [104] Yoichi Katayama, Toshiaki Kitsuki, Yasushi OOI, A Block Processing Unit in a Single-Chip MPEG-2 Video Encoder LSI, Journal of VLSI Signal Processing, 22 (1999) 59-64.
- [105] Yasushi OOI et al, An MPEG-2 Encoder Architecture Based on a Single-Chip Dedicated LSI with a Control MPU, IEEE International Conference On Acoustics, Speech and Signal Processing (ICASSP-97), April 1997, 1 (1997) 599-602.
- [106] Hao-Chieh Chang, Yung-Chi Chang, Yi-Chu Wang, Wei-Ming Chao, Liang-Gee, VLSI Architecture Design of MPEG-4 Shape Coding, IEEE Transactions on Circuits and Systems for Video Technology, 12(9) (2002) 741-751.
- [107] Bontae Koo, Seongmin Kim, Seokho Lee, Minseok Choi, Kihyuk Park, Nakwoong Eum, Jongdae Kim, Hyunmook Cho, A Pipelined Low-Power Architectural MPEG-4 Video Codec Chip with Deblocking Filter for Mobile Wireless Multimedia Applications, Proceedings. 5th International Conference On ASIC, October 2003, 2 (2003) 934-937.
- [108] Kazem Sohraby, Daniel Minoli, Taieb Znati, Wireless Sensor Networks: Technology, Protocols and Applications, A John Wiley & Sons, Inc., Publication, 2007.
- [109] Medetonhan Shambhalla Eugène William, Conception d'une architecture hiérarchique de réseau de capteurs pour le stockage et la compression de données, Université des sciences et techniques de l'université de Franche-Comté, Mars 2010.

LISTE DES PUBLICATIONS

- Ahmed Chefi, Adel Soudani, Gilles Sicard, Hardware compression solution based on HWT for low power image transmission in WSNs, ICM' 2011, 19-22 Dec. 2011.
- Ahmed Chefi, Adel Soudani, Gilles Sicard, A CMOS image sensor with low-complexity video compression for wireless sensor networks, New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International, pp.1,4, 16-19 June 2013.
- Ahmed Chefi, Adel Soudani, Gilles Sicard, Contribution to the design of a CMOS image sensor with low-complexity video compression for wireless sensor networks, Journal of Systems Architecture, Volume 59, Issue 10, November 2013, Pages 818-825.
- Ahmed Chefi, Adel Soudani, Gilles Sicard, Hardware compression scheme based on low complexity arithmetic encoding for low power image transmission over WSNs, AEU - International Journal of Electronics and Communications, Volume 68, Issue 3, March 2014, Pages 193-200.

Résumé:

Ce travail de recherche vise à concevoir un système de vision à faible consommation d'énergie pour les réseaux de capteurs sans fil. L'imageur en question doit respecter les contraintes spécifiques des applications multimédias pour les réseaux de capteurs de vision sans fil. En effet, de par sa nature, une application multimédia impose un traitement intensif au niveau du nœud et un nombre considérable de paquets à échanger à travers le lien radio, et par conséquent beaucoup d'énergie à consommer. Une solution évidente pour diminuer la quantité de données transmises, et donc la durée de vie du réseau, est de compresser les images avant de les transmettre. Néanmoins, les contraintes strictes des nœuds du réseau rendent inefficace en pratique l'exécution des algorithmes de compression standards (JPEG, JPEG2000, MJPEG, MPEG, H264, etc.). Le système de vision à concevoir doit donc intégrer des techniques de compression d'image à la fois efficaces et à faible complexité. Une attention particulière doit être prise en compte en vue de satisfaire au mieux le compromis "*Consommation énergétique - Qualité de Service (QoS)*".

Mots clés: Réseaux de capteurs sans fil, Imageurs CMOS, Compression d'image, faible consommation.

Abstract:

This research aims to develop a vision system with low energy consumption for Wireless Sensor Networks (WSNs). The imager in question must meet the specific requirements of multimedia applications for Wireless Vision Sensor Networks. Indeed, a multimedia application requires intensive computation at the node and a considerable number of packets to be exchanged through the transceiver, and therefore consumes a lot of energy. An obvious solution to reduce the amount of transmitted data is to compress the images before sending them over WSN nodes. However, the severe constraints of nodes make ineffective in practice the implementation of standard compression algorithms (JPEG, JPEG2000, MJPEG, MPEG, H264, etc.). Desired vision system must integrate image compression techniques that are both effective and with low-complexity. Particular attention should be taken into consideration in order to best satisfy the compromise "*Energy Consumption - Quality of Service (QoS)*".

Keywords: Wireless sensor networks, CMOS Imagers, Image compression, Low power.